UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II
SCUOLA DI DOTTORATO IN
INGEGNERIA INFORMATICA ed AUTOMATICA
DIPARTIMENTO DI INGEGNERIA EELETTRICA E TECNOLOGIE DELL'
INFORMAZIONE

# Mechatronics design of a robotic systems

**Luca Rosario Buonocore**

lucarosario.buonocore@unina.it

In Partial Fulfillment of the Requirements for the Degree of
PHILOSOPHIAE DOCTOR in
Computer Science and Automation Engineering

April 2015

TUTORs
Dr. Vincenzo Lippiello
Dr. Sabato Manfredi

COORDINATOR
Prof. Francesco Garofalo

UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

SCUOLA DI DOTTORATO IN
INGEGNERIA INFORMATICA ED AUTOMATICA

Date: **April 2015**

Author: **Luca Rosario Buonocore**

Title: **Mechatronics design of a robotic systems**

Department: **DIPARTIMENTO DI INGEGNERIA EELETTRICA E TECNOLOGIE DELL' INFORMAZIONE**

Degree: **PHILOSOPHIAE DOCTOR**

Permission is herewith granted to university to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

Signature of Author

THE AUTHOR RESERVES OTHER PUBLICATION RIGHTS, AND NEITHER THE THESIS NOR EXTENSIVE EXTRACTS FROM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT THE AUTHOR'S WRITTEN PERMISSION.

THE AUTHOR ATTESTS THAT PERMISSION HAS BEEN OBTAINED FOR THE USE OF ANY COPYRIGHTED MATERIAL APPEARING IN THIS THESIS (OTHER THAN BRIEF EXCERPTS REQUIRING ONLY PROPER ACKNOWLEDGEMENT IN SCHOLARLY WRITING) AND THAT ALL SUCH USE IS CLEARLY ACKNOWLEDGED.

*.... to my parents, to my sister and to Francesca*

*"If you have an apple and I have an apple and we exchange these apples then you and I will still each have one apple. But if you have an idea and I have an idea and we exchange these ideas, then each of us will have two ideas" - George Bernard Shaw*

# Acknowledgments

Finally...after three years...here I am: at a conclusion of a fantastic experience for which I have many people to thank.

First, I deeply thank my advisor Dr. Vincenzo Lippiello. Since my master thesis, his patience, kindness and invaluable support have been the key point for my professional growth. He has been my reference point in this whole new world for me and thanks to his passion and devotion for the research I could move on in the most difficult moments.

I am thankful also to Prof. Luigi Villani whose scrupulousness and advices helped me a lot in the development of my researches.

I have to say a huge "*thank you*" also to Prof. Bruno Sicliano: I discovered the "*robotic world*" through his classes. Moreover, he made possible this Ph.D experience and all my travelings all around the world.

A key point in my life has been represented by the period I spent at Northwestern University under the supervision of Prof. Kevin Lynch. In this period I grew up both from a personal than from a professional point of view. For this, I have to deeply thank Prof.Lynch: his patience, kindness, passion and experience helped me a lot beyond the "*simple*" research activities.

Further, I have to mention all LIMS people who made me feel as at home.

I have to thank also my "local colleagues" of the PRISMA group. They enjoyed with me a lot of time also out of the "standard" situations and working days.

I am also grateful to my Doctorate School, in the persons of Prof. Luigi Pietro Cordella first and then Prof. Francesco Garofalo.

I thank also all the professors, young colleagues, students and friends met during these years in summer schools, European projects meetings and in the robotics conferences and seminars.

My family: initially they did not believe this experience could be useful for me, but during these three years they changed their minds. Nevertheless, I've never felt missing their support, and this allowed me not to give up.

I am thankful also to all my friends for the nice time spent with them: they distracted me when it was useful to restore and fortify myself. Mention all of them in these few lines is impossible...but they know!

Last but absolutely not least, the biggest "*thank you*" (...if this does not hurt

other people) goes to Rita. Her love, patience and support have been essential for the (hopefully) good results in my research works. Of course, she got angry several times for the week-ends I had to spend studying, but with understanding and love all can be come overcome.

*Grazie*!

*The research leading to these results has been supported by the DEXMART Large-scale integrating project, which has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-216239. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.*

Napoli, March 2015

Luca Rosario Buonocore

# Summary

Year by year, service robotics is becoming an attractive field and a cutting-edge research topic, and it has been able to attract the attention of several international research groups. In such a scenario, the European project ArCas (FP7-ICT-287617), under which this thesis has been mainly developed, intends to propose new solutions about bi-manual control in the so called "human-friendly" applications.

Smart collaboration between Humans and ground end aerial Robots for improving rescuing activities in Alpine environments

This work was supported by the Network for Excellence of research programme MASTRI funded by POR Campania FSE 2007-2013. The research leading to these results has been also supported by the ARCAS and SHERPA collaborative projects, which both have received funding from the European Community Seventh Framework Programme (FP7/2007-2013) under grant agreements ICT-287617 and ICT-600958, respectively.

Operating in unstructured environments for such tasks as grasping and manipulation of unknown objects in a "human-like" fashion, fits quite well in the above concerns: a robot, to be truly a part of service robotics applications about the control of aerial application, it should be able to perform autonomous works with or without the presence of humans in the scene.

In this thesis, new architecture of robot arm to grasp and to manipulate unknown objects in different scenario, aerial or ground application.

Then, it is straightforward to recognize that two main tasks have to be performed in such a context: grasping for aerial application and grasp planning for a omini directional base whit a robotic arm. The motivation behind the proposed device provided of sensor for a safety human interaction. The same coordination aspects are the key-points of the last proposed method about comunication: now, the robotic device have to cooperate in order to manage the object in the desired way with a good sensitivity and this can be done exploiting the torque end force sensor of the new system.

In detail, the outline of this thesis is organized as follows.

- Chapter 1 is a general introduction underlining the need to make a robot

autonomous or at least able to operate in unstructured scenarios to cope with human end other robotic device.

- Chapter 2 Project of ultralight robot arm for aerial manipulation whit simulations and experiments.
- Chapter 3 proposed robot manipulator framework in order to exploit the interaction robot-robot, redundancy management for dexterity scopes. Simulations are presented to validate the proposed hardware.
- Chapter 4 exploit the problematic of the comunication and control of the multiple platform. Simulations are presented to validate the proposed approach.
- Chapter 5 contains conclusion, remarks and proposals for possible developments.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Over the course of centuries, human beings tried to design and to build new machines first to help themselves in the execution of several tasks, and then to completely replace themselves, especially in the most dangerous works. In a short time, this desire about having machines in substitution of human being in physical activities has been caught up as well by the desire to substitute him in decision making tasks.

The term *robot* derives from the term *robota* which means executive labour in Slav languages. As well, *robotics* is commonly defined as the science studying the *intelligent connection between perception and action* [49]. These two last definitions show how perfectly a robot fits into the above human being's desires: these last, besides, can be accomplished if and only if the *three fundamental laws* introduced by Asimov are respected. These laws established rules of behavior to consider as specifications for the design of a robot, and they are namely:

- A robot may not injure a human being or, through inaction, allow a human being to come to harm.
- A robot must obey the orders given by human beings, except when such orders would conflict with the first law.
- A robot must protect its own existence, as long as such protection does not conflict with the first and the second law.

Nowadays, robots are widely used in industrial applications for such works where human being would have more risk for his life, more cost per hour and more stress for his body. The connotation of a robot for industrial applications is that of operating in a *structured environment* whose geometrical characteristics are mostly known a priori.

Hence, operating in scarcely structured or *unstructured environments* – where the geometrical characteristics are not known a priori –, even with or in cooperation

with humans, it is not possible without robots with a marked characteristics of autonomy. The expression *advanced robotics* usually refers to this framework, in which the ability in decision making tasks plays a relevant role. Advanced robotics is still a young discipline and therein several researchers are motivated to investigate solutions which could be the answer to the growing need of autonomous robots for domestic and service applications, but also for new industrial requests.

Therefore, robotic systems of the next decade will be, potentially, a part of everyday life as helpers and eldercare companions, assisting surgeons in medical operations, intervening in hazardous or life-critical environments for search and rescue operations and so on. Personal and service robots will thus be found in all domains of our future life, and they represent not only a hope for a more convenient world but also a massive new market for leading-edge technology industry and significant business opportunities, especially for industries. Only a few of the technologies required to build functional personal and service robots already exist at the component level and markets for these products are getting gradually into place. Continuous research and development efforts are required to combine the different technologies, create new products and services, enhance the existing ones for a wide range of possible applications.

The realization of a truly dexterous and autonomous manipulation system is still an open research issue: grasp and manipulation of objects with a robotic quadrotor are such complex tasks combining different strategies constraints, goals, advanced sensing and actuating technologies, requiring new concepts and design of new robotic platforms thanks to new technologies of rapid prototyping (3D printing)

## 1.1 Arcas project

The ARCAS project proposes the development and experimental validation of the first cooperative free-flying robot system for assembly and structure construction.

ARCAS will provide integrated and consolidated scientific foundations for flying robot perception, planning and control. In particular, ARCAS will produce a framework for the design and development of cooperating flying robots for assembly operations.

The integration of these functionalities will pave the way for new applications and services in aerial and space robotics. The building of platforms for the evacuation of people in rescue operations, the installation of platforms in uneven terrains for landing of manned and unmanned VTOL aircrafts, the cooperative inspection and maintenance and the construction of structures, are some examples of aerial robotics' potential.

The project will pave the way for a large number of applications including the building of platforms for evacuation of people or landing aircrafts, the inspection

and maintenance of facilities and the construction of structures in inaccessible sites and in the space.

The detailed scientific and technological objectives are:

- New methods for motion control of a free-flying robot with mounted manipulator in contact with a grasped object as well as for coordinated control of multiple cooperating flying robots with manipulators in contact with the same object. (e.g. for precise placement or joint manipulation).
- New flying robot perception methods to model, identify and recognize the scenario and to be used for the guidance in the assembly operation, including fast generation of 3D models, aerial 3D SLAM, 3D tracking and cooperative perception
- New methods for the cooperative assembly planning and structure construction by means of multiple flying robots with application to inspection and maintenance activities
- Strategies for operator assistance, including visual and force feedback, in manipulation tasks involving multiple cooperating flying robots.

Hence, it is clear from the Summary that this work thesis will be especially focused on the just mentioned firs aspect, in particular to the design of robotic platforms able to manipulate objects during the fly.

The achievement of the research objectives within ARCAS will have an important impact toward the achievements of robust and versatile behavior of artificial systems in open-ended environments providing intelligent response in unforseen situation.

The project will be implemented by a high-quality consortium whose partners have already demonstrated the cooperative transportation by aerial robots as well as high performance cooperative ground manipulation.

## 1.2 Sherpa project

Smart collaboration between Humans and ground and aerial Robots for improving rescuing activities in Alpine environments

Integrated Project FP7-ICT (IP Grant Agreement no.: 600958)

# Chapter 2

# Ultralight Robot Arm for Aerial Manipulation

Grasping and manipulation tasks, in general, require a priori knowledge about the geometric characteristics of the objects and of the scene. The design and implementation of a five degrees-of-freedom robot manipulator conceived for aerial robotics applications is presented in this paper. The robot, named *Prisma Ultra-Lightweight 5 ARm* (PUL5AR), is employed to execute manipulation tasks equipped on board a vertical take-off and landing unmanned aerial vehicle. The arm is ultra-lightweight and its mechanics are thought to decouple its kinematics from the vehicle base roll and pitch rotations and to retain the ability to take-off in little space. Moreover, the mechanical design is conceived to confine the center of gravity of the arm as close as possible to the vehicle frame, thus reducing the total inertia and undesired static torques that should be balanced by the vehicle propellers. The experimental validation of the dynamic model and of several control schemes concludes this work.

## 2.1 Introduction

Vertical Take-off and Landing (VToL) Unmanned Aerial Vehicles (UAVs) are starting to assume increasing importance in civil and military applications. They are being used in tasks involving passive interaction with the external environment, such as inspection, remote sensing and surveillance [15, 16]. The possibility of equipping a UAV with a robotic arm allows it to be used for applications such as grasping and manipulation [45]. However, this scenario poses several new problems. Among all, the dynamic models of the robotic arm [50, 48] and of the UAV [41] should be well estimated and explicitly employed into the control laws to guarantee the stability of the whole system [7]. In fact, the coupling between the two subsystems could cause stability problems, due to both the free

motion of the manipulator, that can be dealt with algorithms estimating unmodeled forces/torques acting on the UAV [47], and the interaction with the environment [46].

In [32] it has been shown that by adding the degrees-of-freedom (DoFs) of the arm to those of the UAV the structural redundancy of the overall system can be fully exploited by hierarchically assigning a number of subtasks.

In some works, aerial pick-and-place tasks have been considered [18, 30], where a few DoFs end effector is made up by a task-specific kinematical structure. The study [25] considers the grasping problem by means of a 3 DoF planar robotic manipulator actuated by servomotors, while in [42] a UAV endowed with two grippers closes a valve. In [22] an unmanned helicopter has been endowed with a KUKA lightweight industrial manipulator; visual measurement and an impedance control algorithm have been employed to execute a grasping task.

A drive train optimization method for the design of light-weight robots is proposed in [53]. In details, an optimal selection of motors and gearboxes from commercially available components is done and the characteristics of the motor and gearbox are considered in the drive train modelling. In [28] a design methodology based on the distributed actuation principle to achieve a high-performance robot manipulator is presented. In fact, spatial movement of the actuation points provides several benefits such as high payload capacity, high efficiency and a lightweight structures.

In this work the problems associated with kinematic coupling between the UAV attitude and the structure of the robot manipulator is addressed by the design of an appropriate differential joint at the base of the robotic arm. Moreover, through the delocalization of the motors at the arm base, the center of gravity (CoG) of the robot arm remains close to the vehicle body frame. In this way, the destabilizing effects on the UAV attitude due to the static torque generated by the misalignment of the CoGs of the UAV and of the arm (see [44]) are significantly reduced. Finally, the ability of interaction with the environment is made possible by using motors and custom electronics to achieve current control (i.e. motor-side torque control), allowing one to implement impedance control algorithms such as those described in [31].

This work also takes into account all the constraints related to aerial robotics such as payload and limited energy storage [26]. In fact, the UAV maximum payload shall be considered: it has to be high enough for the aerial robot to carry the manipulator, its electronics, sensors and actuators and eventually a grasped object (task payload). This also increases the amount of energy necessary to execute tasks, i.e. it reduces the flight autonomy. Hence the need for lightweight robot manipulators, which must also take into account the challenges and limitations described before. In this paper we also discuss how the choice of the actuators, the mechanical design based on 3D printing, and other factors such as the transmission of motion, affect the ability of the arm in terms of both payload and manipulation

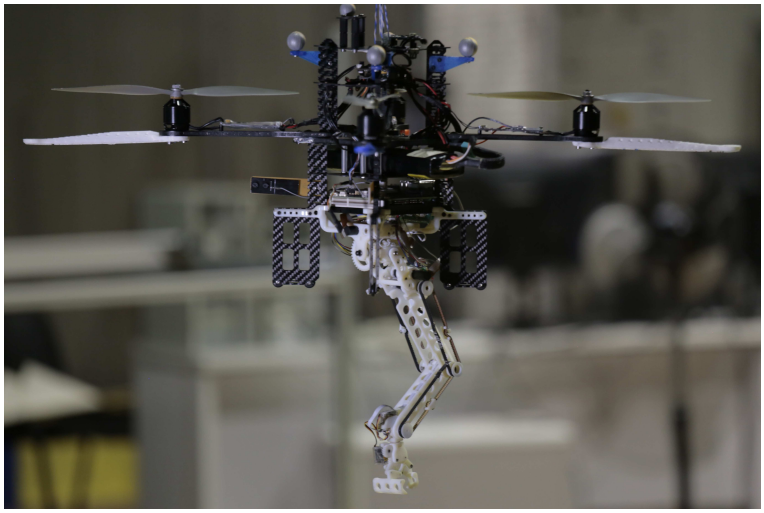


Figure 2.1: ASCTEC PELICAN quadrotor equipped with the PUL5AR robotic arm.

capability of the transported payload. The mechanical design of the arm is also constrained by the opportunity to fold onto itself, in order to allow the UAV to take-off and land without the need of particular equipment and by reducing the total inertia of the system during free flight. The target UAV used during the design is an ASCTEC PELICAN (see Fig. 2.1), but the arm can be easily mounted on any other VToL UAV with similar payload and whose controller is equipped with a USB port.

Experimentation has been carried out by using several control modules and communication software specifically written for the arm.

## 2.2 Mechanics

The mechanical design of the PUL5AR robot arm has been inspired by the specifications resulting from the final application, i.e. aerial manipulation. The target UAV is an ASCTEC PELICAN [1] with a payload of 650 g, but similar vehicles can be considered as well. In details, the considered constraints are as follows:

- Arm weight: 250 g.
- Arm payload: 200 g.
- Arm maximum extension: 300 mm.
- Retaining the possibility to execute VToL and reduce the total inertia during free flight.

- Decoupling of flight kinematics and dynamics.

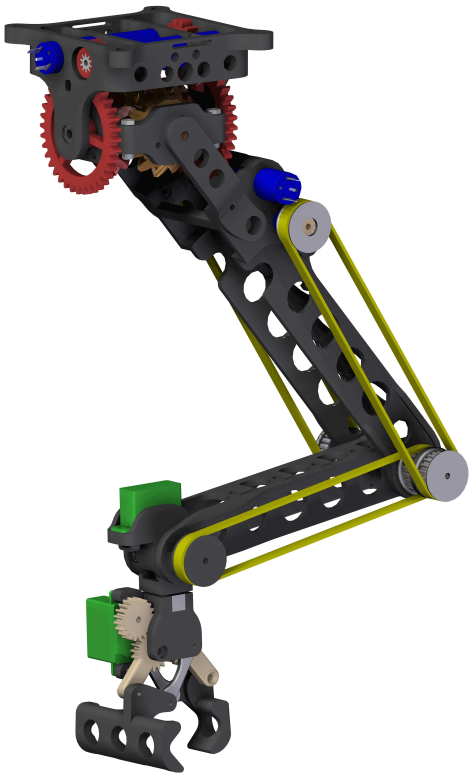


Figure 2.2: CAD model of the PUL5AR robot arm.

The arm maximum weight has placed a set of limits mostly on the mechanical design but also on the choice of the motors and electronic components. Notice how the ratio between the arm weight and payload is 1.25 unlike typical industrial robotic arms for which this ratio is about 20. The designed CAD model of the PUL5AR robot arm is shown in Fig. 2.22, where the use of a honeycomb structure is highlighted to reduce the total weight. However, this kind of choice influences the total elasticity of the mechanical structure, hence a suitable FEM analysis has been carried out during the design phase (see Fig. 2.23).All the mechanical parts have been obtained using an OBJet24 3D printer which was able to realize objects with a very high resolution using a material composed by acrylic monomer (see Fig. 2.1).

The manipulation tasks for which the PUL5AR robot arm is designed suggests mounting it on the lower part of the UAV, tampering with its VToL ability. Therefore, the structure was designed to be able to fold onto itself completely,

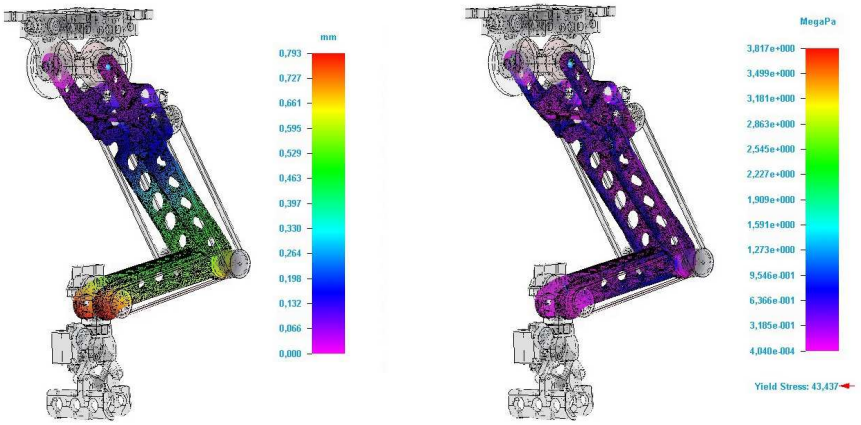



Figure 2.3: FEM analysis: displacement (left) and stress (right) due to a weight of 160 g acting on the gripper.

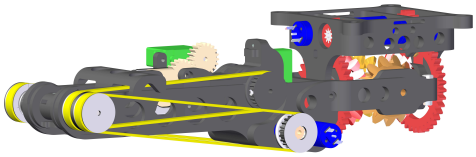


Figure 2.4: CAD model of the PUL5AR robot arm in folded configuration.

hence reducing to the minimum space occupied by the arm during the take-off and landing phases as shown in Fig. 2.24.

Figure 2.5: The designed belt fastener: the screw exerts a force on the timing belt.

The arm has been designed with a total of six actuators: 4 direct current (DC) brushed motors (for the first four joints) and 2 digital servomotors (for the last joint and the gripper). The first two DC motors are mounted on the base of the arm, the third and fourth are mounted at the beginning of the second joint, away from the joints they actuate. For the latter, the motion is transmitted by means of MXL timing belts 4.8 mm wide. In order to make it easier to mount the belts and transmit the motion more efficiently, we have designed a *belt fastener* whose aim

is to fasten the belts by means of the pressure applied by a screw (see Fig. 2.25).

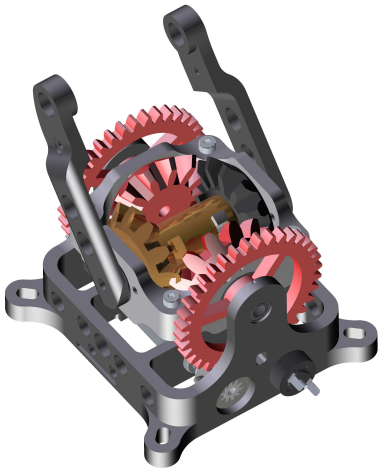


Figure 2.6: Close up of the U joint.

The first two joints have been designed in order to allow a direct compensation the effects of the UAV attitude variation on the manipulator end-effector position. Hence, the base joint has been designed as a differential joint (see Fig. 2.6), which can mostly compensate the UAV pitch and roll rotations. Moreover, this kind of structure has allowed mounting the first two motors inside the base, reducing even more the arm's inertia. Further, the adoption of timing belts allowed to delocalize the other two DC motors at the base of the first arm link. The disadvantage is that of having more backlash due to the differential and transmission mechanisms.

The CoG of the arm is thus confined to be close to the base frame. In Fig. 2.7 the position of the arm CoG $\boldsymbol{p}_{CoG}(\boldsymbol{q})$ with respect to different joint configurations $\boldsymbol{q}$ corresponding to the gripper center $\boldsymbol{p}_e(\boldsymbol{q})$ moving along the vertical and horizontal axes are shown. Figure 2.8 highlights how the CoG moves away from the base slowly with respect to the corresponding distance of the gripper when the latter is far from the base. This behavior is particularly advantageous because it allows moving the arm in all the workspace by introducing a limited static disturbance on the UAV body.

The total arm size is constrained to be mounted on the lower side of a UAV, an ASCTEC PELICAN in our case study. The arm's maximum reachable distance and the maximum payload are a design tradeoff constrained by the electrical and mechanical power of the actuators and gearboxes. In view of the considered specifications, the MAXON RE 10 motors have been chosen. They have a weight/power factor of 12 g/W, with a maximum continuous torque of 1.2 Nm joint side. The reduced dimension of the structure has suggested the use of two digital servomotors to actuate the last joint and the gripper, as shown in Fig. 2.26. However,

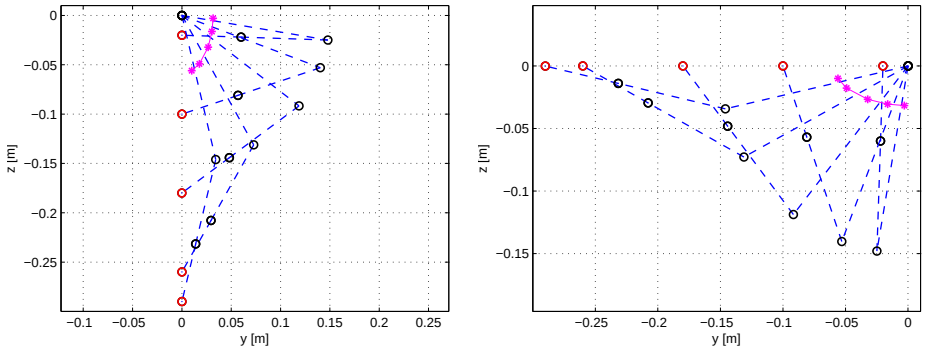



Figure 2.7: Variation of the position of the arm CoG (magenta) with respect to different joint configurations corresponding to the gripper center (red) moving along the vertical (left) and horizontal (right) axes.

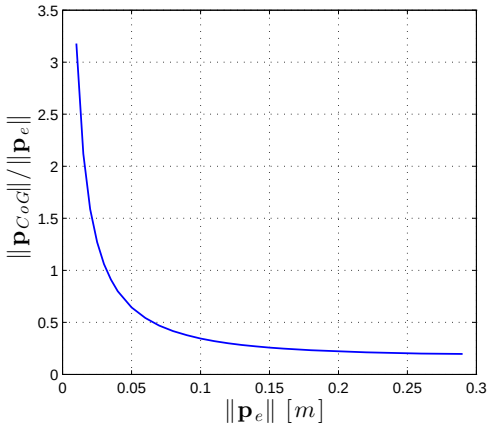



Figure 2.8: Diagram of the ratio of the distances of the robot CoG $\boldsymbol{p}^b_{CoG}$ and of the gripper position $\boldsymbol{p}^b_e$, from the robot base, with respect to $\|\boldsymbol{p}^b_e\|$.

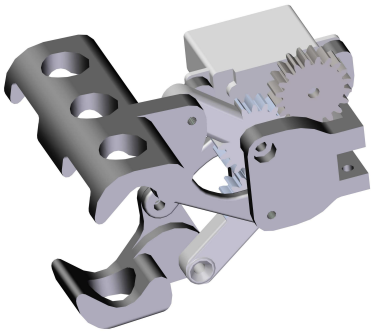


Figure 2.9: The CAD model of the gripper.

having DC motors for the first four joints, i.e. the joints mainly responsible for the positioning of the end-effector, enables the implementation various interaction control schemes, among which impedance control, noticeably.

## 2.3 Modeling

The model of the PUL5AR robot arm has been derived following the formulation given in [50]. Kinematic modeling is based on the choice of a set of reference frames according to the Denavit-Hartenberg convention. During this phase the mechanical coupling existing between joints has also been considered. Dynamics are obtained using the Euler-Lagrange energy formulation. Finally, the symbolic math formulation has been derived using the robotics toolbox described in [9].

### 2.3.1 Reference frames, kinematic coupling

The Denavit-Hartenberg (DH) convention represents a set of rules to conveniently choose a set of reference frames that can describe the kinematics of the chain of rigid bodies. A possible choice is briefly described in Table 2.1 and in Fig. 2.10.

| Link | $a_i$ | $\alpha_i$ | $d_i$ | $\vartheta_i$ |
|---|---|---|---|---|
| 1 | 0 | $\pi/2$ | 0 | $\vartheta_1$ |
| 2 | 150 | 0 | 0 | $\vartheta_2$ |
| 3 | 88 | -$\pi$ | 0 | $\vartheta_3$ |
| 4 | 0 | $\pi/2$ | 0 | $\vartheta_4$ |
| 5 | 0 | 0 | -60 | $\vartheta_5$ |

Table 2.1: Denavit-Hartenberg parameters (lengths are in mm).

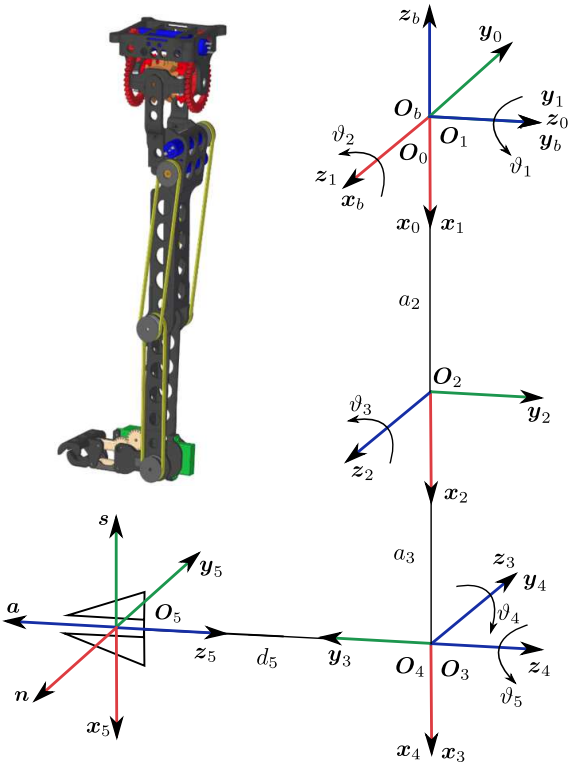



Figure 2.10: The joints reference frames chosen following the Denavit-Hartenberg convention. The robot is shown in $\boldsymbol{q} = \boldsymbol{0}$ configuration.

The vector of the generalized coordinates is given by the independent variables found in the DH table, i.e. $\boldsymbol{q} = \begin{bmatrix} q_1 & \cdots & q_5 \end{bmatrix}^T = \begin{bmatrix} \vartheta_1 & \cdots & \vartheta_5 \end{bmatrix}^T$ for the PUL5AR robot arm.

The kinematic coupling existing between the joints, which is due to the presence of the differential joint and of motion transmission mechanisms, can be represented in compact form as follows:

$$\boldsymbol{q}_m = \begin{bmatrix} k_r k_d & k_r k_d & 0 & 0 & 0 \\ -k_r k_d & k_r k_d & 0 & 0 & 0 \\ 0 & 0 & k_r & 0 & 0 \\ 0 & 0 & -k_r & k_r & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{q} = \boldsymbol{H}\boldsymbol{q}, \tag{2.1}$$

where the vector $\boldsymbol{q}_m \in \mathbb{R}^5$ defines the rotation of each motor, while $k_r = 256$ and $k_d = 3$ are the chosen gear ratio and the differential gear ratio, respectively. The matrix $\boldsymbol{H}$ projects a joint space configuration to the motor space configuration. It can be easily shown that $det(\boldsymbol{H}) = 2k_d^2 k_r^4 > 0$, hence (2.1) can be inverted

yielding:

$$\boldsymbol{q} = \boldsymbol{H}^{-1}\boldsymbol{q}_m. \tag{2.2}$$

Equations (2.1) and (2.2) show that $q_{m1}$ and $q_{m2}$ actuate only joints $q_1$ and $q_2$, while $q_{m3}$ and $q_{m4}$ actuate joints $q_3$ and $q_4$. Joint $q_5$ is independent from the others because it is directly actuated by a servomotor with no transmission mechanisms other than a gear ratio, as is the case with the gripper, which is not described here since it does not contribute to the kinematic model. Since $\boldsymbol{H}$ is constant, one can differentiate (2.1) yielding[1]:

$$\dot{\boldsymbol{q}}_m = \boldsymbol{H}\dot{\boldsymbol{q}}. \tag{2.3}$$

The relationship between the motor torques $\boldsymbol{\tau}_m$ and the joint torques $\boldsymbol{\tau}$ can also be found. Assuming an ideal model for the gear ratios, the following relation holds:

$$\boldsymbol{\tau}^T\dot{\boldsymbol{q}} = \boldsymbol{\tau}_m^T\dot{\boldsymbol{q}}_m = \boldsymbol{\tau}_m^T\boldsymbol{H}\dot{\boldsymbol{q}} \quad \forall\dot{\boldsymbol{q}} \quad \Rightarrow \quad \boldsymbol{\tau}_m = \boldsymbol{H}^{-T}\boldsymbol{\tau}. \tag{2.4}$$

### 2.3.2 Direct and differential kinematics

The direct kinematics of a robot manipulator can be compactly represented by a homogeneous transformation matrix as follows:

$$\boldsymbol{T}_e^b(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{R}_e^b(\boldsymbol{q}) & \boldsymbol{p}_e^b(\boldsymbol{q}) \\ \boldsymbol{0} & 1 \end{bmatrix}, \tag{2.5}$$

where $\boldsymbol{R}_e^b(\boldsymbol{q})$ and $\boldsymbol{p}_e^b(\boldsymbol{q})$ represent the rotation matrix and the position of the gripper with respect to the base frame, respectively, as a function of the robot configuration $\boldsymbol{q}$. On the other hand, the differential kinematics provides the relationships existing between velocities in the joint space and the end-effector linear and angular ones in Cartesian space. This relationship is linear and is represented by the so-called *geometric Jacobian* matrix (see [50]). One has:

$$\boldsymbol{v}_e = \begin{bmatrix} \dot{\boldsymbol{p}}_e(\boldsymbol{q}) \\ \boldsymbol{\omega}_e(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_P(\boldsymbol{q}) \\ \boldsymbol{J}_O(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{2.6}$$

The analytical expressions of $\boldsymbol{T}_e^b(\boldsymbol{q})$ and of $\boldsymbol{J}(\boldsymbol{q})$ are here omitted for brevity.

Notice that, since the Cartesian DoFs are 5, the corresponding geometric Jacobian is a $(6 \times 5)$ matrix. In fact, the orientation of the end effector cannot be completely specified if its positions has already been assigned. This means that the trajectory planner has to take into account the presence of this constraint. This also correspond to choosing an *operational space* with a dimension lower or

[1] Due to the kinematic coupling, the transformation between $\dot{\boldsymbol{q}}$ and $\dot{\boldsymbol{q}}_m$ is not really linear, but should be modified to take into account the saturation on the maximum velocity of each motor, which does not linearly map onto velocity saturations on the joints, but makes $\boldsymbol{H}$ configuration dependent.

equal to the DoFs in the Cartesian space. A possible choice for the orientation can be:

$$\boldsymbol{\theta}(\boldsymbol{q}) = \begin{bmatrix} \phi(\boldsymbol{q}) \\ \psi(\boldsymbol{q}) \end{bmatrix} = \begin{bmatrix} q_2 + q_3 - q_4 \\ q_5 \end{bmatrix}. \tag{2.7}$$

In fact, this choice turns out to be natural one by considering that the manipulator mechanical structure lies entirely, at any time, on a plane if the position $\boldsymbol{p}_e^b(q)$ has been specified. Hence, it is natural to assign the rotation of the end effector along the gripper approaching axis and $\boldsymbol{z}_1^b(\boldsymbol{q})$.

Notice that, a standard VToL UAV has 4 DoFs, hence by adding a manipulator with other 5 DoFs, a redundant system is achieved, even if it is not the case of the arm alone.

### 2.3.3 Kinematic singularities

By inverting (2.6), the velocities in the operational space are projected into the joint space via the generalized inverse of matrix $\boldsymbol{J}(\boldsymbol{q})$. Whenever $\text{rank}(\boldsymbol{J}(\boldsymbol{q})) < 5$, the equations in (2.6) become linearly dependent. The joint space configurations $\boldsymbol{q}$ which correspond to this condition are termed *singular configurations*. From these, a solution $\dot{\boldsymbol{q}}$ can be obtained only if $\boldsymbol{v}_e \in \mathcal{R}(\boldsymbol{J})$. In this case, the motion is locally physically executable, even if the configuration is singular. If instead it is $\boldsymbol{v}_e \notin \mathcal{R}(\boldsymbol{J})$, the system of equations in (2.6) has no solution and the motion is not executable. The analysis of the existence of singular configurations is crucial when using closed-loop inverse kinematics algorithms that make use of the inverse of the analytical Jacobian matrix. The evaluation of the determinant of the analytical Jacobian of the PUL5AR robot arm yields:

$$\begin{aligned} \det(\boldsymbol{J}_A(\boldsymbol{q})) = \frac{1}{2}(a_2 a_3 (a_2 sin(\vartheta_2 + \vartheta_3) + d_5 cos(\vartheta_2 + 2\vartheta_3 - \vartheta_4) - a_3 sin(\vartheta_2) + \\ - d_5 cos(\vartheta_2 - \vartheta_4) - a_2 sin(\vartheta_2 - \vartheta_3) + a_3 sin(\vartheta_2 + 2\vartheta_3))). \end{aligned} \tag{2.8}$$

One can observe that $\det(\boldsymbol{J}_A(\boldsymbol{q}))$ does not depend on $\vartheta_1$ nor on $\vartheta_5$, while it can be shown that:

$$\vartheta_3 = 0 \implies \det(\boldsymbol{J}_A(\boldsymbol{q})) = 0 \ \ \forall \vartheta_1, \vartheta_2, \vartheta_4, \vartheta_5. \tag{2.9}$$

### 2.3.4 Dynamics

The dynamic model of the PUL5AR robot arm has been obtained using the Lagrange approach [50]. By recalling that the arm has $n = 5$ DoFs, the equations of motion can be conveniently written as:

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}_v\dot{\boldsymbol{q}} + \boldsymbol{F}_s sgn(\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{H}^T\boldsymbol{\tau}_m - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}_e \tag{2.10}$$

where $\boldsymbol{B}(\boldsymbol{q})$ is the positive definite and symmetric inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ models the centrifugal and Coriolis effect, $\boldsymbol{g}(\boldsymbol{q})$ is the gravity term, $\boldsymbol{F}_v$ and $\boldsymbol{F}_s$ are diagonal and positive definite matrices representing viscous and static friction effects, respectively, and $\boldsymbol{h}_e$ is the vector of generalized forces exerted by the end effector on the environment. These equations of motion are function of a set of dynamic parameters. In particular, for each link $i$ one can define:

- mass $m_{\ell_i}$;
- inertia tensor $\boldsymbol{I}_{\ell_i} \in \mathbb{R}^{3\times3}$;
- center of mass $\boldsymbol{p}_{\ell_i}$.

In order to simplify the model, the mass of the motors and their inertia tensor are included in the link related parameters. In particular, masses $m_{\ell_5}$ and $m_{\ell_4}$ respectively includes the masses of the servomotors $m_{m_6}$ and $m_{m_5}$ (where $m_{m_6}$ is the mass of the gripper actuator). The parameter $m_{\ell_2}$ takes into account masses $m_{m_3}$ and $m_{m_4}$. Parameters $m_{m_1}$ and $m_{m_2}$ do not give any contribute to the equations of motion, since the first two motors are mounted on the base (these two quantities have to be added to the mass of the UAV).

All the inertial parameters have been estimated through the CAD model and the motor data-sheets of the motors: Table 2.2 (2.3) shows the mass and CoG (inertia tensor elements) of each link.

| Link | $m$ | $p_{CoG_x}$ | $p_{CoG_y}$ | $p_{CoG_z}$ |
|---|---|---|---|---|
| 1 | 28.654 | 0.00 | -0.02 | 0.87 |
| 2 | 73.16 | -86.37 | 10.4 | 0.23 |
| 3 | 11.815 | -41.44 | 1.71 | -1.42 |
| 4 | 7.968 | -3.93 | 0.11 | 8.15 |
| 5 | 14.52 | -6.88 | -3.86 | 28.73 |

Table 2.2: Mass (g) and CoG (mm) of each link. The CoG $\boldsymbol{p}_{C_i}$ of link $i$ is expressed in the $i$th frame.

| Link | $I_{xx}$ | $I_{xy}$ | $I_{xz}$ | $I_{yy}$ | $I_{yz}$ | $I_{zz}$ |
|---|---|---|---|---|---|---|
| 1 | 96.20 | 0.01 | 0.00 | 49.48 | 0.00 | 66.08 |
| 2 | 219.96 | -106.84 | 26.85 | 1298.95 | -3.18 | 1147.79 |
| 3 | 11.23 | -0.80 | -11.00 | 161.04 | 0.32 | 156.05 |
| 4 | 7.24 | 0.04 | -1.27 | 9.32 | -0.08 | 3.76 |
| 5 | 37.23 | 2.46 | 0.68 | 44.65 | -3.82 | 25.61 |

Table 2.3: Inertia tensor elements ($10^{-7}\mathrm{kg}\cdot\mathrm{m}^2$) of each link. Inertia tensor $\boldsymbol{I}_i$ of link $i$ is expressed with respect to a frame attached to its CoG and rotated as the $i$th frame.

Notice that, for each link $i = 1, \ldots, 5$, the quantities $\boldsymbol{p}_{\ell_i}$ and $\boldsymbol{I}_{\ell_i}$ are constant with respect to a reference frame with the origin in $\boldsymbol{p}_{\ell_i}$ and orientation as the $i$th frame.

### 2.3.5 Dynamic model validation

The experimental validation of the CAD measurement has been done by comparing the measured torques acting on the motors (more precisely the corresponding currents) with the ones predicted by the dynamic model. In our case, an indirect torque measurement $\boldsymbol{\tau}$ can be obtained from the measured currents using:

$$\boldsymbol{\tau} = \boldsymbol{H}^T \boldsymbol{\tau}_m = \boldsymbol{H}^T \boldsymbol{k}_t \boldsymbol{i}_a, \tag{2.11}$$

where $\boldsymbol{k}_t \in \mathcal{R}^{4\times4}$ is the diagonal matrix of motor current constants, and $\boldsymbol{i}_a \in \mathcal{R}^4$ is the vector of measured motor currents. The evaluation of the predicted torques $\bar{\boldsymbol{\tau}}$ also needs the accelerations $\ddot{\boldsymbol{q}}$ (see (2.10)). We obtained them by first filtering the velocities $\dot{\boldsymbol{q}}$ using a mean average filter with a fixed-size sample window, and then by numerical differentiation.

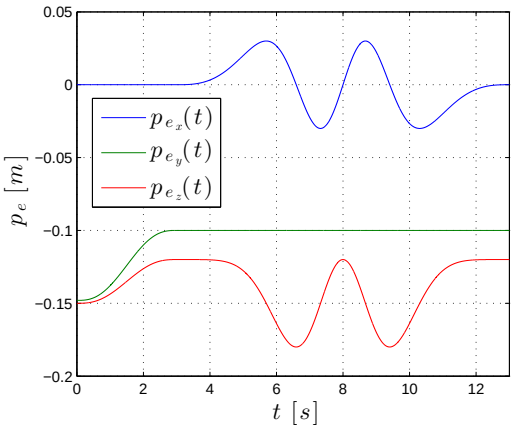



Figure 2.11: Time history of the reference trajectory in the Cartesian space employed for the dynamic model validation.

The estimation errors corresponding to the Cartesian motion of Fig. 2.11 are shown in Fig. 2.12, where one can notice that, by using a highly accurate 3D printer and by estimating friction through a suitable identification process, the parameters estimated from the CAD model allow obtaining an accurate model.

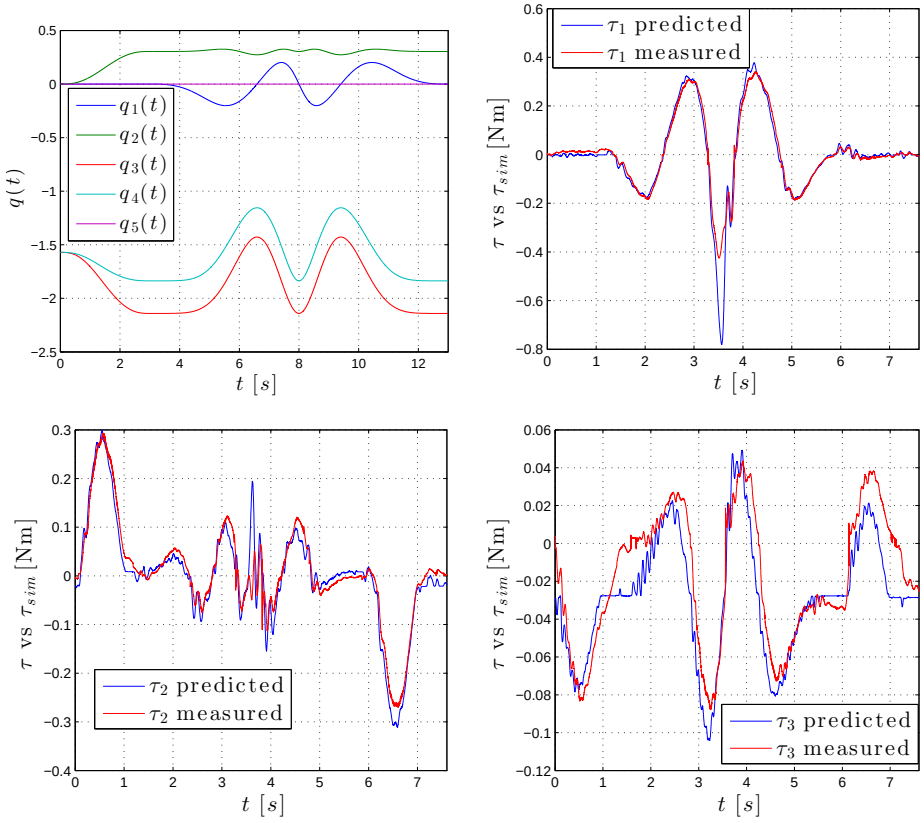



Figure 2.12: Time history of the joint trajectories employed for the dynamic model validation (top-left), and dynamic model validation: comparison between the predicted (blue) and the measured (red) values of $\tau_1$ (top-right), $\tau_2$ (bottom-left), and $\tau_3$ (bottom-right).

## 2.4 Custom Electronics

Limitations on the total weight of the arm affect not only mechanics, but also the electronics of the robot. A single microcontroller has been adopted as an interface between the communication layer and the direct access to the electronical components (motor drives, sensors, etc.). The chosen microcontroller is the MBED LPC1768 [3].

The use of a centralized host controller establishes some problems on the communication of the data with each motor controller. The matter gains even more importance since we wish to close the feedback loops at high frequencies (4 kHz for the current loop, 1 kHz for the velocity and position loop) to make possible

the implementation of a high-level interaction control scheme.

For each DC motor the acquisition of the position and current is required. Therefore, we chose to drive data through a *Serial Peripheral Interface* (SPI) bus, which allows a high data rate and a limited number of connections. This also allows sampling the data as close as possible to the sources, limiting electrical noise.

One of the main reasons for using the bus with dedicated measurement acquisition chips is that this lightens the microcontroller computational load, since the electronical components and the bus takes care of low-level tasks, such as hardware interrupt management due to the encoder signals. The microcontroller has to implement the communication protocol through the USB and read the data on the SPI, in order to realize the current, velocity and position feedback control loops.

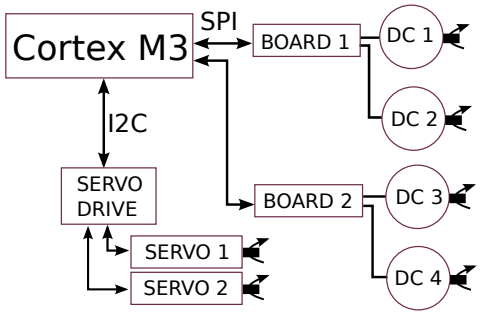



Figure 2.13: Hardware architecture of the low-level control system.

In order to further lighten the computational load, the servomotors are interfaced by means of an integrated circuit (a PCA9685 that operates on the I2C bus) that allows them to send appropriate PWM control signals. The overall architecture of the designed electronic board is shown in Fig. 2.13.

The DC motors are driven through a phase and a 40 kHz PWM signal, obtained directly from the microcontroller. The motor drive is a Texas Instruments DRV8801 Full Bridge able to supply 2.8 A peak current, enough to manage the stall currents of the selected motors. These drives also supply a measure of the current flowing through the motor, which is also amplified and filtered on the power drive and acquired by an ADC MCP3202, which is able to acquire 100 ksample/s. The current is finally digitalized on the SPI bus and read by the microcontroller. The encoder pulses are counted by a dedicated chip, an LS7366. This is a 32-bit counter working at 40 MHz acquiring frequency. The block scheme of the power drive and acquisition modules is shown in Fig. 2.14.

Since the selected motors are endowed with an incremental encoder, it has been necessary to devise a way to sense the proximity of mechanical joint limits.

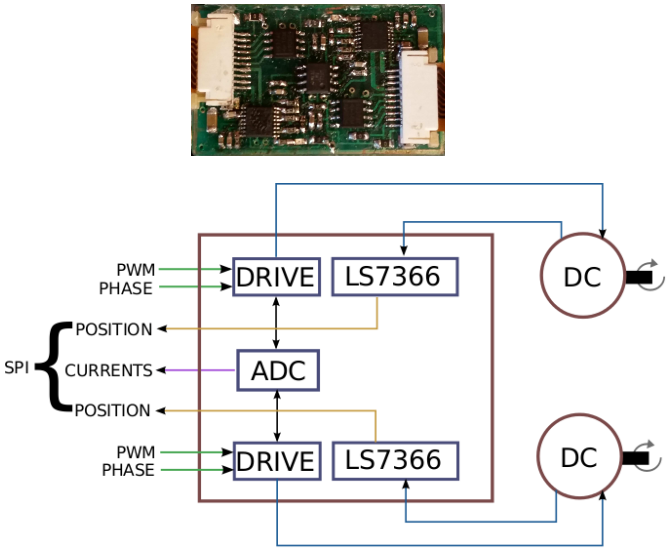



Figure 2.14: Block scheme of the power drive and acquisition modules (named "BOARD#" in Fig. 2.13).

Hence, we have used mechanical switches acting as joint limits detectors. These have offered the possibility to implement a *homing* procedure, allowing the arm to find its folded configuration and kinematically calibrate itself in a known position. The switches are directly connected to the microcontroller.

## 2.5 Communication Protocol

A software framework and a communication protocol for the robot to be commanded by a host computer has been developed. The latter could be any device equipped with a USB port.

In order to properly interpret a data packet exchanged between the motor microcontroller and the host, a communication protocol has been designed and implemented. The protocol provides the possibility to specifying a command, an identifier, and a data field. The host can request a reading, send a reference value or set the configuration of the microcontroller or of the connected devices. In this mode of operation, the host (master) synchronizes the microcontroller (slave), which continuously waits for a new request. A second mode of operation has also been designed, in which instead it is the microcontroller that continuously sends the sensor readings and triggers the host. The host then replies with a packet containing reference values for the motors. This mode is useful for control loops in order to synchronize and have the latest sensor readings at the start of a control

cycle without having to request them explicitly.

The protocol has been designed in order to have dynamically sized packets to optimize bandwidth usage. Each data packet (see Fig. 2.15) has a 2-byte header and a variable length data field. The header is made up by:

- $C \rightarrow$ command byte: it specifies how to correctly interpret the packet data field.
- $I \rightarrow$ identification byte: it specifies to which object (DC motor, servomotor, feedback loop, etc.) the command has to be addressed.


Figure 2.15: Structure of a generic data packet: command (C), identification (I), and data field. Each square represents a byte.

The designed microcontroller firmware implements the command and identification bytes definitions in a single header file, using a list of predefined commands, associating a value of dimensions *uint8_t* (1 byte long) to each of them. Given how a packet is constructed and indicating with $n$ the number of values converted to floats in the data field, a given packet will be $2 + 4n$ bytes long.

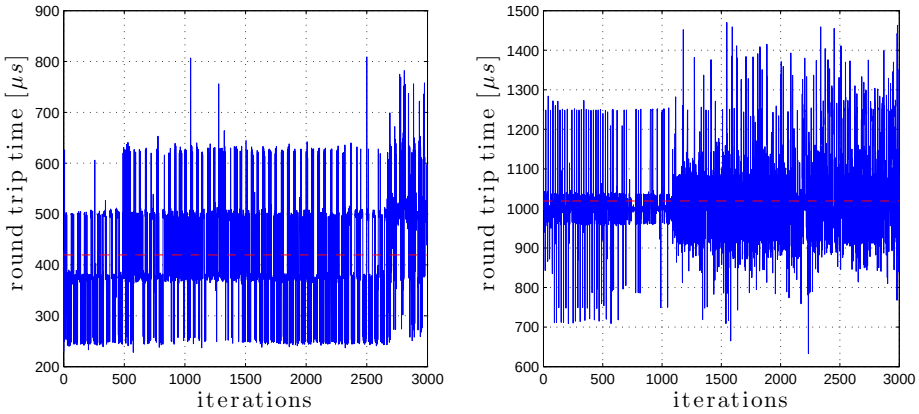



Figure 2.16: RTT measured for the request of all the motor positions (left: mean value is about 420 $\mu$s) and for sending and receiving a 50 byte packet (right: mean value is about 420 $\mu$s).

The communication protocol relies on the USB protocol as a means to physically exchange the data packets. We have chosen this low-level protocol both

because it has a sufficiently high bandwidth for high-speed data transmission and it makes it easy to quickly connect the robot to any computer equipped with a USB port.

In order to validate the choice, the *RTT* or *round trip time* —the time delay between the start of a request packet and the end of the reply— of the average data packet has been measured. In Fig. 2.16 the RTT measured for the request of all the motor positions and the mean RTT for sending and receiving a 50 byte packet are shown. The achieved results confirm that a control loop of more than 500 Hz can be implemented with a reasonable safety margin.

One must consider that we currently use bulk USB transmissions, which guarantees integrity of the packet but not the arrival delay. This is one of the reasons that generates the variation of the communication time, as shown in Fig. 2.16. To further improve the stability of the communication timing, work is in progress on the possible use of isochronous USB transmission. This kind of communication guarantees a bounded time delay on data transmission, but not on the arrival of all the data packets, like instead with the UDP protocol. Thus, in order to exploit the bounded delay offered by the asynchronous transmission and implement a more stable control loop, it could be necessary to implement bare data exchange checking.

The high-level library software runs on the host and hides the details of the communication implementation. The user can set the arm parameters (such as configuration, reference signals, etc.) by using the functions available in the library.

## 2.6 Control

Both independent joint control and centralized control algorithms have been implemented and tested on the developed robot arm: PD with gravitational compensation, inverse dynamics, cascade control. In the following some experimental results achieved with these control techniques are presented to evaluate the performance of the designed robot arm. The Cartesian reference trajectory of Fig. 2.11 is employed in all tests.

### 2.6.1 PD with gravitational compensation

The employed PD with gravitational commendation (PD+) control scheme is shown in Fig. 2.17, where $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are positive definite gain matrices (see [50]). It is based on gravity compensation, a proportional action on the error in joint space, and a derivative action based on the actual joint velocity estimation.

Figure **??** (left) shows the time history of the norm of the tracking error. The amplitude of the error is always less than 1 mm, but one can notice a residual error of about 0.5 mm at the steady-state, the latter mainly depending on an imperfect cancellation of gravity term $\boldsymbol{g}(\boldsymbol{q})$ due to the dynamic calibration errors. In fact,

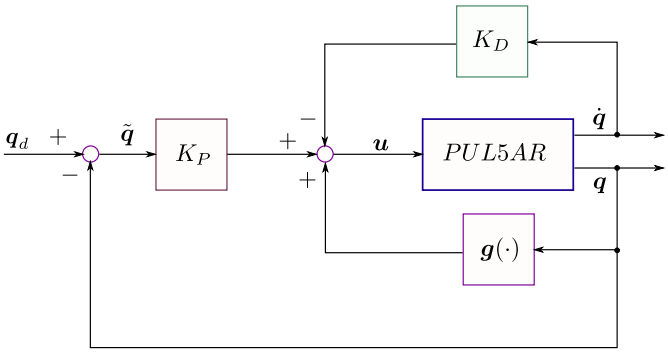



Figure 2.17: PD+ control block scheme.

the error is mainly concentrated along the $z$-axis of the inertial frame, as shown in Fig. **??** (right). The introduction of a robust or an integral action can be employed

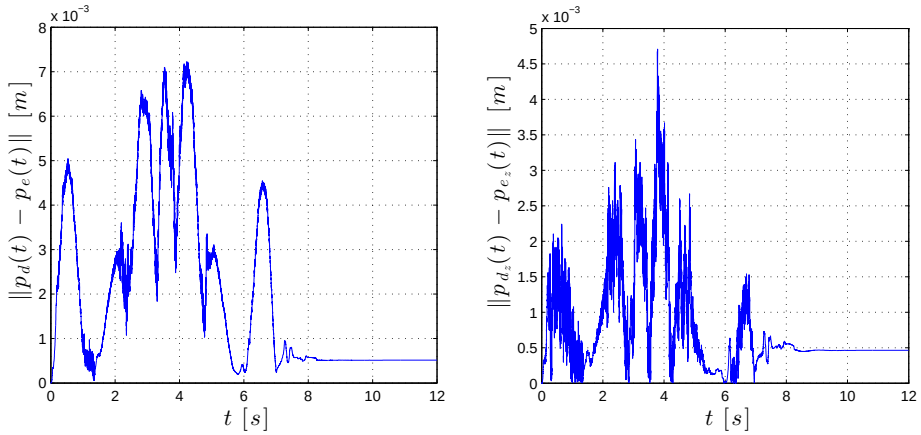



Figure 2.18: PD+ control scheme: position error in norm (left) and along the $z$-axis of the inertial frame (right) of the trajectory shown in Fig. 2.11.

to eliminate the steady-state error.

### 2.6.2 Inverse dynamics

An inverse dynamics control scheme is based on the complete cancellation of the dynamic model. The control block scheme is represented in Fig. 2.19, where $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q})$, and $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are positive definite gain matrices. Notice that the cancellation of the nonlinear dynamics of the robotic system relies on the perfect knowledge of the dynamical parameters. The norm of the achieved position tracking error is shown in Fig. 2.20 for which similar considera-

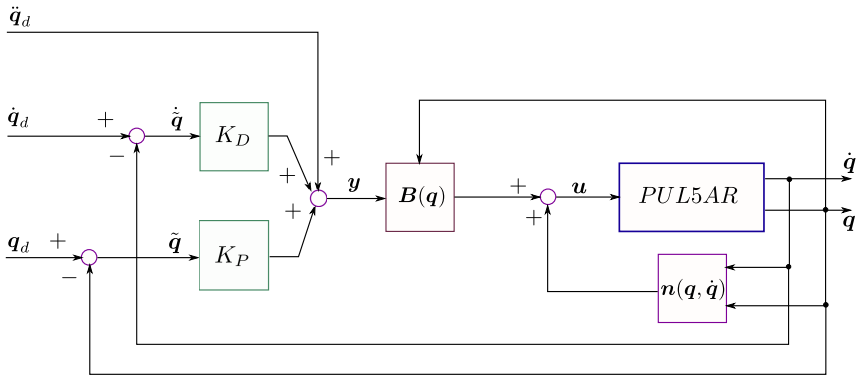



Figure 2.19: Inverse Dynamics control block scheme.

tions on the imperfect compensation of the gravitational term as for the previous scheme can be done.

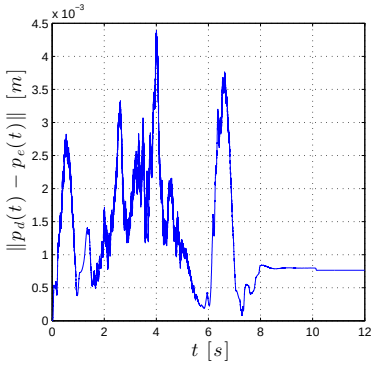



Figure 2.20: Inverse dynamics control scheme: position error in norm of the trajectory shown in Fig. 2.11..

### 2.6.3 Cascade control

A cascade loop of current, velocity and position controllers has been implemented and tested on the proposed setup. The current and velocity loop contain proportional and integral actions, while the position loop can also consist of the proportional action only. Design of the control gains typically goes from the inner towards the outer loop. Feedforward velocity and torque references (evaluated through the direct dynamic model) can also be provided, hence enhancing dynamic performance.

The achieved tracking error in norm is shown in Fig. 2.21. It can be observed

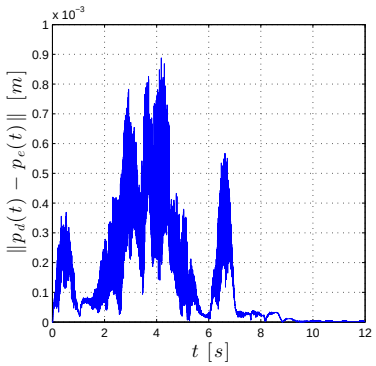



Figure 2.21: Norm of tracking error with the trajectory shown in Fig. 2.11.

that the error is less than 1 mm during the transient and goes to zero when the manipulator reaches the final pose thanks to the presence of an integral action in the control loop.

## 2.7 Second Version of Robot Arm for Aerial Manipulation

### 2.7.1 Mechanics

The mechanical design of the PUL5AR robot arm has been inspired by the specifications resulting from the final application, i.e. aerial manipulation. The target UAV is an ASCTEC PELICAN [1] with a payload of 650 g, but similar vehicles can be considered as well. In details, the considered constraints are as follows:

- Arm weight: 900 g.
- Arm payload: 500 g.
- Arm maximum extension: 600 mm.
- Retaining the possibility to execute VToL and reduce the total inertia during free flight.
- Decoupling of flight kinematics and dynamics.

The arm maximum weight has placed a set of limits mostly on the mechanical design but also on the choice of the motors and electronic components. Notice how the ratio between the arm weight and payload is 1.25 unlike typical industrial robotic arms for which this ratio is about 20. The designed CAD model of the PUL5AR robot arm is shown in Fig. 2.22, where the use of a honeycomb structure

Figure 2.22: CAD model of new version of robot arm.

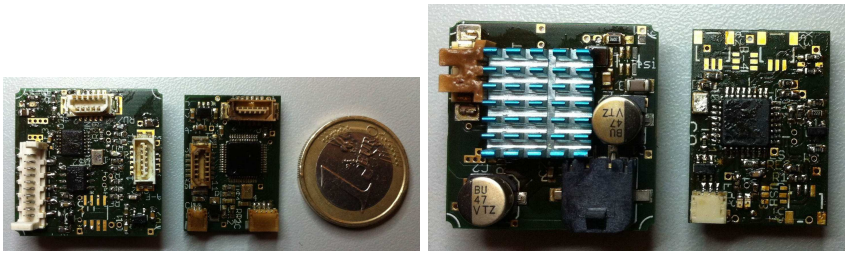


Figure 2.23: Drive end controller photo.

is highlighted to reduce the total weight. However, this kind of choice influences the total elasticity of the mechanical structure, hence a suitable FEM analysis has been carried out during the design phase (see Fig. 2.23).All the mechanical parts have been obtained using an OBJet24 3D printer which was able to realize objects with a very high resolution using a material composed by acrylic monomer (see Fig. 2.1).

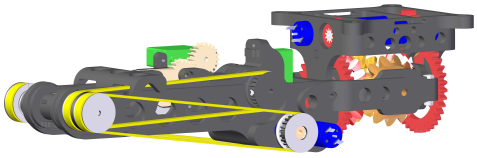


Figure 2.24: CAD model of the PUL5AR robot arm in folded configuration.

The manipulation tasks for which the PUL5AR robot arm is designed suggests mounting it on the lower part of the UAV, tampering with its VToL ability. Therefore, the structure was designed to be able to fold onto itself completely,

hence reducing to the minimum space occupied by the arm during the take-off and landing phases as shown in Fig. 2.24.

Figure 2.25: The designed belt fastener: the screw exerts a force on the timing belt.

The arm has been designed with a total of six actuators: 4 direct current (DC) brushed motors (for the first four joints) and 2 digital servomotors (for the last joint and the gripper).

which can mostly compensate the UAV pitch and roll rotations. Moreover, this kind of structure has allowed mounting the first two motors inside the base, reducing even more the arm's inertia. Further, the adoption of timing belts allowed to delocalize the other two DC motors at the base of the first arm link. The disadvantage is that of having more backlash due to the differential and transmission mechanisms.

This behavior is particularly advantageous because it allows moving the arm in all the workspace by introducing a limited static disturbance on the UAV body.

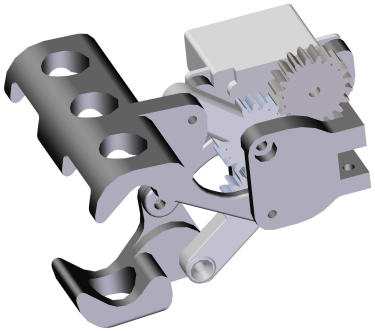


Figure 2.26: The CAD model of the gripper.

The total arm size is constrained to be mounted on the lower side of a UAV, The arm's maximum reachable distance and the maximum payload are a design tradeoff constrained by the electrical and mechanical power of the actuators and gearboxes. In view of the considered specifications, the MAXON RE 10 motors have been chosen. They have a weight/power factor of 12 g/W, with a maximum continuous torque of 1.2 Nm joint side. The reduced dimension of the structure has suggested the use of two digital servomotors to actuate the last joint and the

gripper, as shown in Fig. 2.26. However, having DC motors for the first four joints, i.e. the joints mainly responsible for the positioning of the end-effector, enables the implementation various interaction control schemes, among which impedance control, noticeably.

### 2.7.2 Torque sensor

During this phase the mechanical coupling existing between joints has also been considered. Dynamics are obtained using the Euler-Lagrange energy formulation. Finally, the symbolic math formulation has been derived using the robotics toolbox described .

# Chapter 3

# Mobile platform for grasping

## 3.1 Introduction

In recent years the availability of sensors and hardware ever more efficient and cheap has favored the development of autonomous systems able to move in an environment integrating in an intelligent manner with the same. In parallel, in the industrial field have been developed robotic arms ever more efficient and lightweight can be secured even in environments where there is the presence of the human being. The aim of the present study is to merge these two results by providing the student or researcher a robotic platform able to be performing manipulation tasks in unstructured environments. In particular elaborated on the design of a mobile robotic platform to 8 degrees of freedom formed by a robotic arm with 5-axis gripper integrated and from a mobile base omnidirectional. The platform was equipped with a series of sensors for the kinematics and the interaction with the environment, in particular it has available in addition to the measuring position and speed of the individual joints also a measure of the torque of each joint of the arm and on obtained by each wheel torque sensors designed and built by the candidate. Furthermore, the platform will be equipped with on-board computer and other sensors (Kinect, optical sensors for odometry, IMU) so as to be used for testing algorithms autonomous robotics. In particular, the platform has been made in the context of SHERPA project, of which a brief description will be given later, in order to test the algorithms developed for the terrestrial platform..

### 3.1.1 Mobile robots in unstructured environments

In Workshop held in 2006 at the University of Pennsylvania in Philadelphia [27] challenges are discussed the main challenges facing the autonomous robotics in unstructured environments. In particular, we discussed what were the major tar-

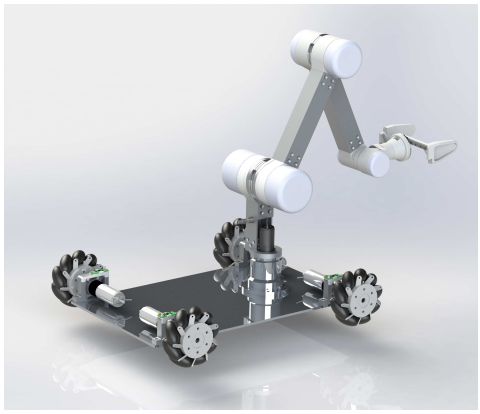


Figure 3.1: Scheme of robotic arm end system of reference

gets of scientific research such as to enable the integration of the robot in different environments from the industrial context, such as the domestic context. Mainly unlike the industrial context, the robotics research in unstructured environments must address a number of issues including:

- Dynamic environment: change the structure of the environment or other properties.
- Uncertainties on the pose of objects to be manipulated, on their shape and size and the presence of obstacles is not known.
- Uncertainty on the interaction of the robot with other robots or humans
- Constraints real-time.
- Issues relating to safety during user interaction with robots

In order to solve the problems listed in the workshop they discussed what were the main objectives of that scientific research should focus, including:

- **Project of robot:** The robot had to be designed already at the hardware level so as to be easily integrated in contexts not structured. Therefore required that:
  - anthropomorphic robots must have characteristics similar to those of the human being (the size of the arms and hand) in order to facilitate his interaction with objects designed for man.
  - The mobile robot should be able to move in complex contexts such as the domestic environment (for example the robot Roobma is built with a low design so as to be able to move easily passing under beds or under furniture.

  - Manipulator arms have to be lightweight while maintaining a high payload in order to ensure the safety of the user. An example is given by robots from Kuka Light weight products.

- **Perception:** Perception is one of the areas in which research has focused more in recent years. In particular the two perceptual organs are most important in a robot::
  - Vision Sensors: With these sensors it is possible to equip the robot's ability to recognize objects to be manipulated and its installation in the space, to recognize the environment and localize in it, to recognize the user. One of the sensors used in the research is the sensor Kinect product from Microsoft that provides a measure of the pose of an object in three dimensional space.
  - tactile and force sensors: With these sensors, the robot is able to interact with the user or with the environment in a dynamic, changing its pliability to the contact with the user, or adapting to environments not known. An example of a robotic hand with tactile sensors, thanks to them, the robot may be able to perform the grasping of objects not known simply measuring the forces exerted on them by the fingers of the hand.

The mobile robots, such as the one designed in the present study, blend the dexterity of a manipulator arm with the flexibility of a mobile base by adding the ability to integrate many types of sensors aimed at the development of algorithms for autonomous robotics. An example is the commercial product Youbot from Kuka (fig 3.2) from which you take the basic idea nevertheless amending certain key aspects such as:

- Double size arm respect product Kuka
- Payload double
- Further dimensions of the base
- Integration of torque sensors in the joints of the arm and the base

### 3.1.2 SHERPA project

one of the objectives of the SHERPA project is to develop a robotic platform mixed (air and ground) (fig 3.3) able to support search and rescue operations in hostile environments like the real mountain scenery.

The project has a strong degree of heterogeneity due to the presence of different agents [37] :

- The rescuer human who is expert in surveillance activities and recovery of missing (mountain guide or ranger). The human wire less transmits its position to the robotic platform and communicates with it via intuitive systems that allow natural interaction without distracting from its activities (voice commands and gestures)


Figure 3.2: Kuka Youbot

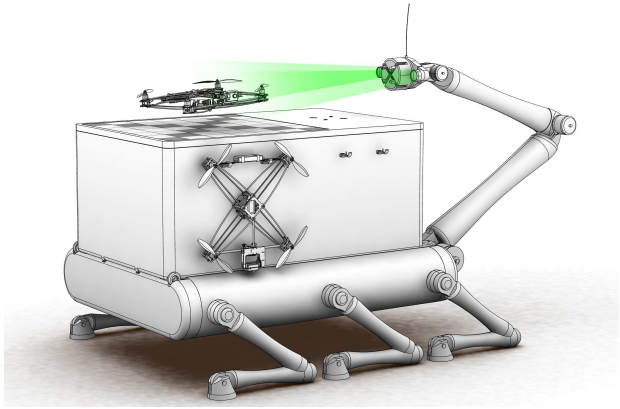


Figure 3.3: Project SHERPA: mobile platform with robotic arm

- Small unmanned aerial vehicles equipped with cameras and sensors used in support of surveillance expanding the area monitored by the human rescuer. Are designed to operate with a high degree of autonomy and be controlled by the user in a simple and natural, also must be able to operate in proximity of humans without harm it. To this are vehicles of reduced dimensions, which of course entails a reduced autonomy and a reduced payload transportable.
- Big planes at high altitudes complete the SHERPA team with complementary characteristics to other UAV: high strength, high payload. They are used to build a 3D map of the area of rescue, as HUB communication among other platforms if the area is morphologically critical to patrol larger areas and in the case to carry all the boxes SHERPA in areas not accessible to the rover.
- A rover of land that serves as a form of transport for the equipment of the

rescuer, as the station hardware with great computational power, and as a charging station for other vehicles. It features multifunctional robotic arm used for handling and to support other vehicles in the take-off and landing (fig 3.4). It is designed to work with a high degree of autonomy to both the cognitive level and duration of the power system. Communicate with other players via wireless robotic and human actor through natural controls.

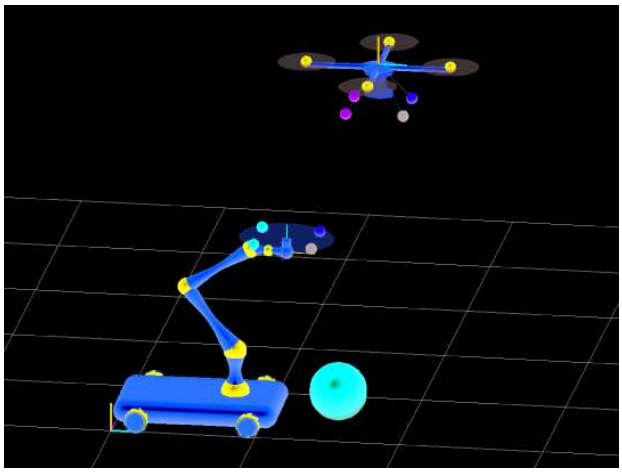


Figure 3.4: Project SHERPA: assisted landing

All agents interact and collaborate using their features and capabilities, toward achieving a common goal. An advanced control system and cognitive abilities characterize the SHERPA system, which aims to support the rescuer improving its awareness of the scene, helping in the process of search for the missing and rescue of the same. The main features of the system are SHERPA:

- Acquisition of cognitive skills.
- Development of collaborative strategies
- Natural interaction and implied between the user and the robot sherpa (through voice, gestures, etc.).

In order to ensure a natural interaction between the actors and the rescuer robotic platforms.

## 3.2 Mechatronics

### 3.2.1 Mechanical Design

It briefly shows the mechanical design activities carried out, indicating mainly the problems faced and their relative solution. Starting from the basic work carried out at the laboratory PRISM lab, which were taken to the guidelines regarding the size of the main mechanical elements, the entire robot was designed using the drawing environment simulation and CAD Solid Works 2015. In particular, there has occupied the engineering of all mechanical elements aimed to a more simple construction with the machines available and the resolution of some issues below.

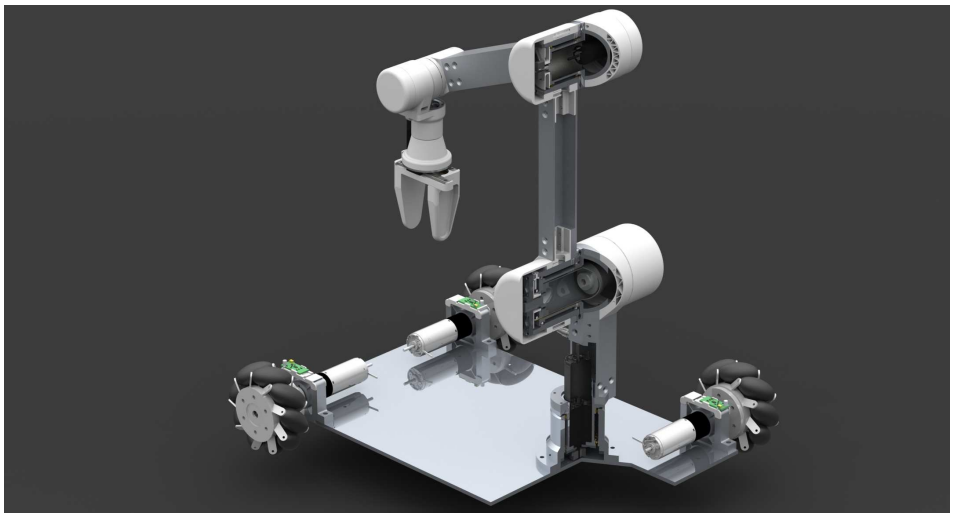


Figure 3.5: CAD Mobile Robot

#### Mobile Base

Design specifications:

- Omnidirectional mobile base.
- Torque sensors capable of measuring the extent of the external forces acting on the base.
- Low center of gravity of the base so as to make the structure more stable during the motion of the arm and during the phase of acceleration and deceleration of the base.
- Contact points of omnidirectional wheels with the floor belonging to the vertices of a square in order to simplify the kinematic model of the base.

implementation:

- Have been used omnidirectional wheels *Meccanum* produced by *Nexus Robot* that provide a good grip even with rapid movements.
- Have been designed, as will be widely discussed below, the torque sensors capable of measuring the couples exerted on the shaft of each motor. However, the direct measurement of the shaft torque of the engine has been impractical due to the presence of the cables for the power supply and the management of the sensor, therefore, was chosen as an alternative to the measurement of the torque exerted by the motor on the basis that, although subject to large disturbances, guaranteed equally good results.
- In order to minimize the overall dimensions in order to make low the center of gravity, the supporting structure has been realized with an aluminium plate of 5mm and the engine mounts are designed to integrate the torque sensors of each wheel.

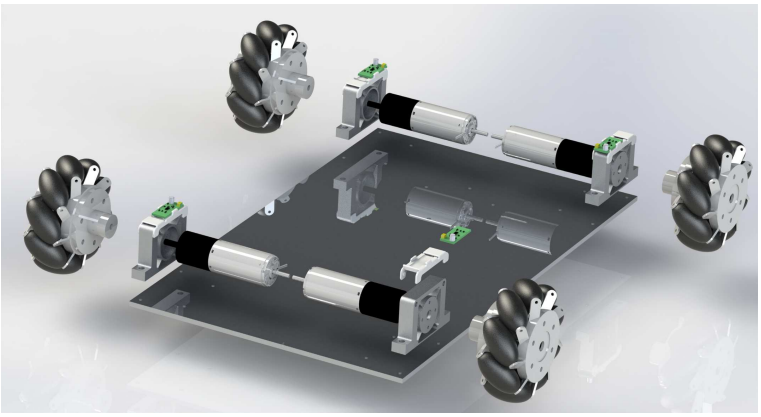


Figure 3.6: exploded base

### Joints of the robotic arm

Design specifications:

- Easy construction of the parts with machinery available.
- Coupling compact torque sensor integrated.
- Control cables and power within the joint.
- Good stiffness in a weight content.

Implementation:

- Starting from the project carried out at the laboratory PRISMA Lab, choices of the main dimensions of the joints (size of the bearings and motors). were dealt with the engineering of the structure of the joint breaking down into simpler components, easily achievable with the machinery available.
- for each coupling is designed a torque sensor perfectly integrated in its geometry and compatible with the range of applicable torque .
- In order to keep the overall dimensions, each joint is designed in such a way as to be integrated inside the engine while maintaining the center of gravity at the center of the joint. This choice, in addition to reducing the size of the single joint, has also simplified the dynamic model of the robot with regard to the contribution of inertia of the rotor. How do you assess the following images (fig3.8 e fig3.9)the mobile part of the joint is integral with the motor and rotates relative to the fixed part by means of two ball bearings, the motor shaft is connected to the fixed part through the torque sensor. This structure has guaranteed, unless problems related to manufacturing tolerances described below, the only measure of torque along the axis of the coupling and isolation of all other forces.
- For the joint two and the coupling tree is a mechanism designed for the passage of cables inside the joint. In this regard it has been exploited the gap between the motor and the joint to pass the cables and has been studied a mechanical element can act both as a closure element of the joint is from winder for cables (fig3.7)in order to guarantee its integrity during the motion of the joint.

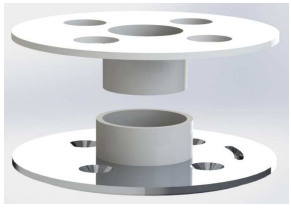


Figure 3.7: Winder cable management

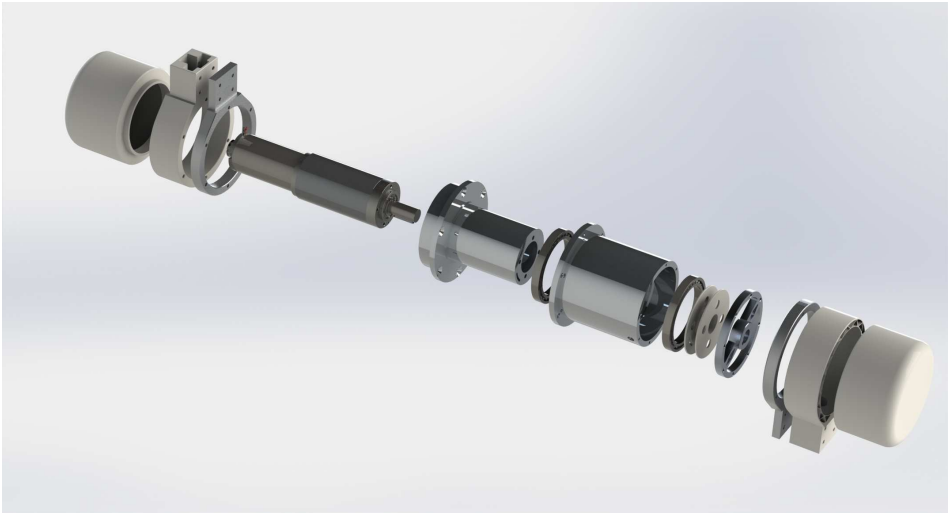


Figure 3.8: Explode joint one

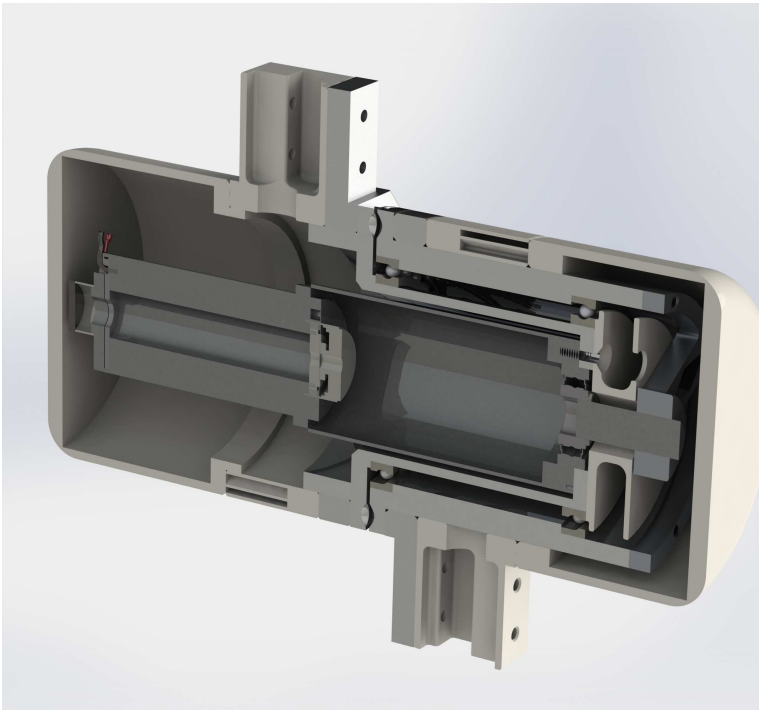


Figure 3.9: Section of joint one

### Structural elements of the robotic arm

Design specifications:

- Weight content.
- Space inside for the cables and for some boards of control and power.
- Good torsional rigidity.

Implementation:

- Were performed FEM analysis of various types of profiles to choose the best one in order to achieve the connecting elements between the various joints of the robot. Were evaluated two hypotheses, the use of a H-profile consists of two C-profiles coupled and the use of a square profile. The first had the opportunity as a convenience to have a space in which to secure easy access control cards and power. However as a result of the analyzes performed (fig3.11) the H-profile has been discarded because of poor torsional rigidity and therefore was chosen profile picture, which guaranteed a stiffness by almost two orders of magnitude higher.
- The anchoring elements of the link between the profiles and the joints are designed so as to ensure a good stiffness at low weight. In particular for the coupling, 0 is chosen to use only the aluminum, for the coupling 1 has been used a mixed solution of aluminum-plastic while for the other joints was used a solution based only on plastic elements. For the plastic elements has also been used a geometry such as to guarantee a good rigidity while reducing the weight and cost of the material(fig3.10). Every choice has been supported by FEM analysis in environment Solid Works.

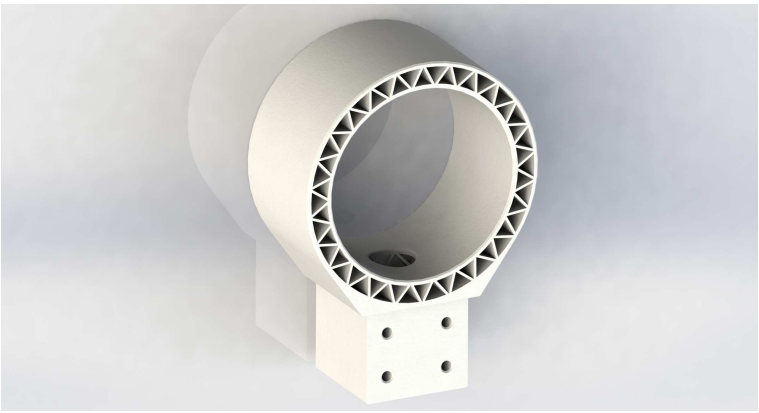
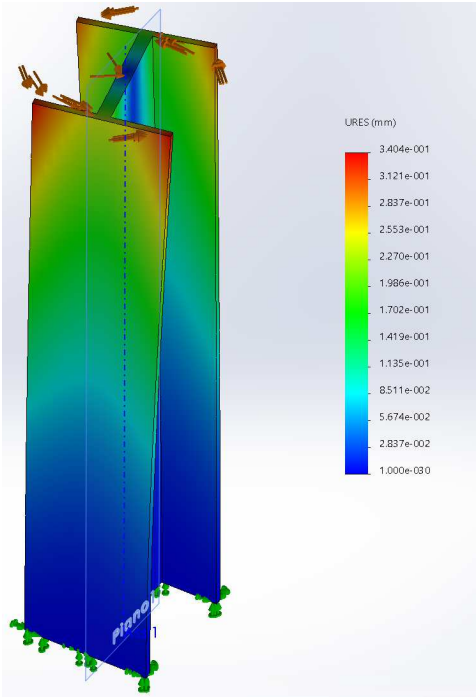
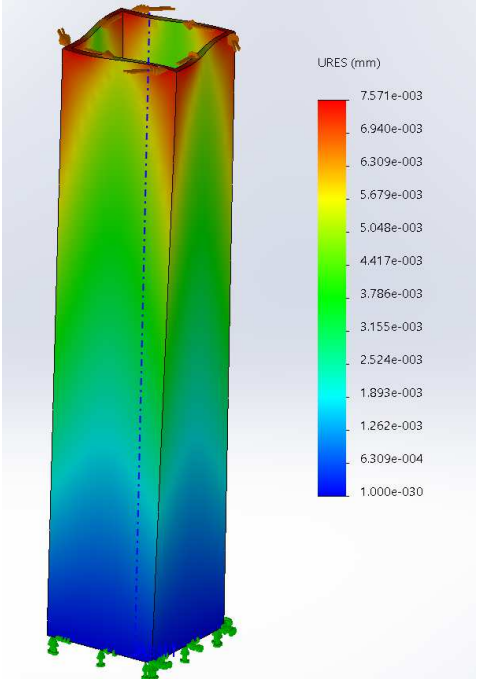


Figure 3.10: Plastic element connecting the joint 2 and the link




(a) *Analysis FEM H-profile(Apply torque 4Nm).*




(b) *Analysis FEM Square-profile (Apply torque 4Nm).*

Figure 3.11: FEM analysis of the profiles for the links

### Gripper (hand effector)

Design specifications:

- Large opening range.
- Must be integrated with the second joint of the wrist so as to minimize the size and therefore the moments exerted on the wrist in order to maximize the pay-load.
- Integration of a torque sensor for the measurement of the efforts of opening and closing of the gripper itself.

Implementation:

- can be observed from the figures, (3.12, fig3.13, fig3.14)has been devised a mechanism for opening and closing of the fingers based on linear recirculating ball handled by a system of gear-racks. This has allowed to greatly reduce the size of the structure and allowed to enter a torque sensor between the gear implemented by the engine to that incident on the racks in order to measure the torque on the latter, and then to reconstruct the forces exerted by 'object on the fingers during the closing phase.
- The whole structure is designed in plastic due to the complexity of the mechanical elements, except for some structural components made of aluminium.

Figure 3.12: Explode of the gripper

Figure 3.13: Open gripper

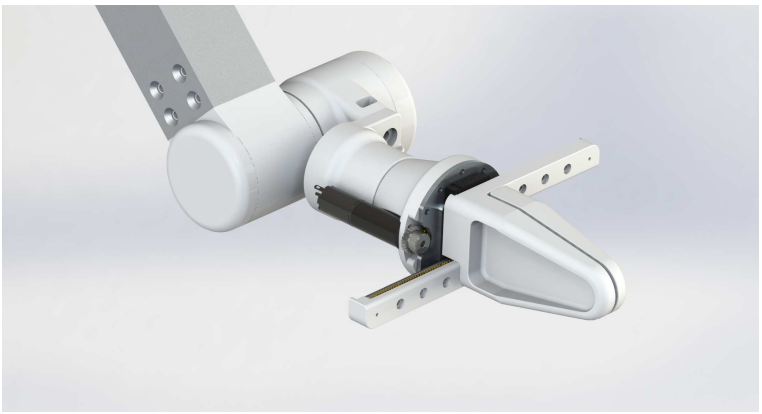


Figure 3.14: Closed gripper

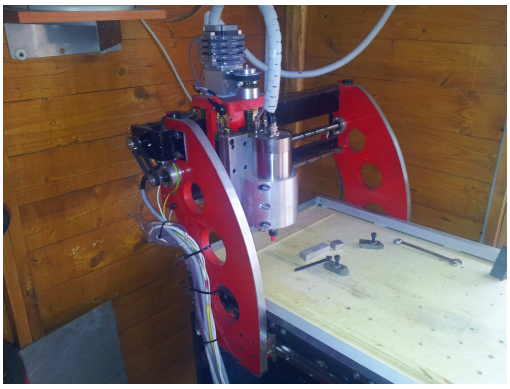
### 3.2.2 Notes on the construction

All mechanical components of the robot were built using machinery and technologies described below. The processes were performed with control machine creating tool paths through the CAD-CAM software CAMWorks. The components electronic was designed and assembled internally

#### Equipment used

Have been used the following equipment during the construction phase of the mechanical details of the robot:

- machinery manuals:
  - Manual lathe
  - Photo engraving machines (bromografo)
- automatic machinery
  - 3D printer (fig3.15-b) Object 24 produced by Stratasys which has the following characteristics:
    - Working area: 240X200X150mm
    - Resolution: 28u
    - Workable material: rigid white (VeroWhitePlus), ideal for modelling and generic prototyping.
  - CNC milling machine (fig3.15-a) reported briefly the main features:
    - Working area: 600X400X200mm
    - Resolution: 0.05mm
    - Repeatability: 0.1mm
    - Workable material: Wood, plastic materials, carbon fibre, aluminium, small operations on steel



(a) *CNC milling.*

(b) *3D printer.*

Figure 3.15: Equipment

#### Assembling

In figure(3.16) shows the image of the drive and the micro controller, instead in Figure (3.17) shows the complete robot.

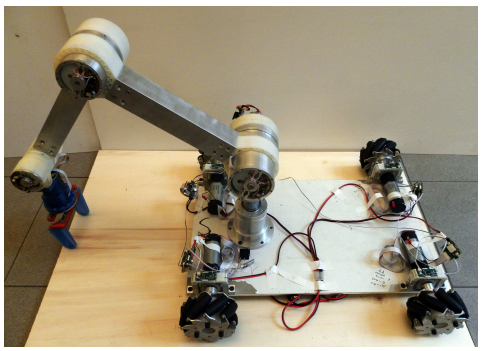


Figure 3.16: Electronic components of contol

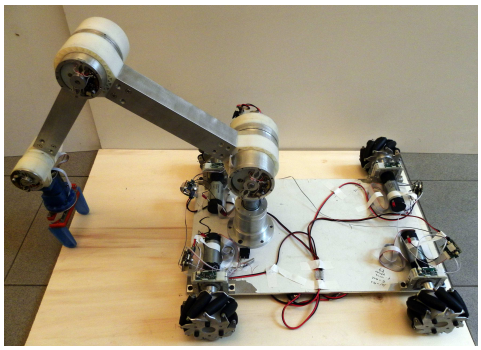


Figure 3.17: Completed robot

### 3.2.3 Torque sensor

The measure of the force acting on a robot manipulator has been one of the most important challenges in the field of robotics research in recent years. In particular, thanks to it was possible to design advanced control algorithms such as:

- Impedance control
- Control of compliance
- Force control and hybrid controls force-position

These controls can be used both in an industrial context to facilitate the assembly and processing (Peg hole in [20]) or (Theaching by showing), or are widely used in research to encourage interaction man- robot. In fact obviously in contexts not structured the robot can not behave as a rigid entity but must be able to adapt its behavior to user behavior in order to ensure the safety by being yieldable on contact with the user or with the environment . The torque measurement on a robot manipulator can be realized mainly in two different ways:

- Measurement of forces generalized agents on end-effector with the aid of a sensor Maltese cross (figure **??**). The advantage is the ease of installation and also on robots not born for the direct measurement of the torque. The main disadvantage concerns the impossibility of exploiting the redundancy of the robot in the controls mentioned above. In fact from the measurement of the forces acting on end-effector is obviously impossible to reconstruct other forces acting on the body of the robot, forces which may be used to realize the internal motions (in the case of redundant robots) intended to make the whole structure yieldable interaction with the user.

  - Indirect measurement of the torque provided by the measure of the motor current $i_A$ of the single joint, from which you can derive the torsion $C$ to the joint by the following relation:

    $$C = K_t K_r i_A \qquad K_t = Reduction\,ratio \qquad K_r = Torque\,Constant \quad (3.1)$$

  - Direct measurement of the torque through torque sensors embedded in each joint. The advantage of course is that unlike the indirect measurement, the measurement by torque sensors is much less noisy and much more linear as in the indirect measurement of non linear factors intervene such as:

    * Torque dead zone
    * Noise due to the switching system of the engine (brushes for DC motors)
    * Noise due to the fuel system of the engine (PWM for DC motors)
    * Noise due to the inertia of the rotor

In order to contain the costs of the platform and to have a greater flexibility in the design has been chosen not to use commercial sensors but to build all the sensors of the robot. In particular, the cost of each sensor (mechanical and acquisition board) was less than 15 euro, much less than the commercial sensors.

**Torque sensors based on strain gauge**

Suppose we want to measure the force acting on a bar (Figure 3.19-b), if the material has a linear behavior, such as a metal, then it will be possible to reconstruct

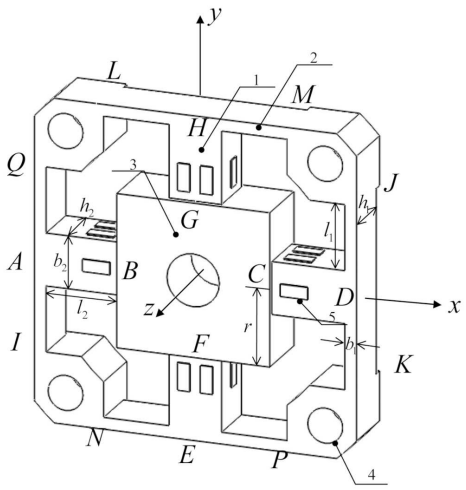



Figure 3.18: Force - Torque Sensor.

such a force by measuring the deformation of the material itself. To this end, in the market are the sensors called Strain Gauge: The strain gauge is a sensor composed of a metal wire with low thermal coefficient inserted in a thin film of plastic material to form a resistive element. Glued on the surface deforming changes its resistance when subjected to tension or compression, and then to vary the deformation of the structure on which it is glued. By designing appropriately the structure is then possible to measure forces or couples simply measuring the resistance change on the strain gauge itself.

We define:

- $\epsilon$ Deformation (Strain)
- $L$ Original length
- $\Delta L$ Stretching
- $E$ Elastic modulus of the material
- $\sigma$ Stress
- $R$ Resistance of Strain Gauge unstretched
- $\Delta R$ Variation of the resistance value
- $K_s$ Gage factor

The following equations bind Stress at Strain and the variation of resistance:

$$\sigma = E\epsilon = E\frac{\Delta L}{L} \tag{3.2}$$

$$\frac{\Delta R}{R} = K_s * \epsilon \tag{3.3}$$

There are various types of Strain Gauge, the differences are:

- **The type of resistive material used:**
  - copper wire and nickel: strain gauge are the cheapest, have a wide range of permissible deformation gage factor but have a relatively low (less than 5)
  - Silicon chip: they are much more expensive, have a range of permissible deformation lower than those based on copper wire, but have a gage factor much larger (about 100). Which is a huge advantage because it requires amplification considerably lower.
  - Polysilicon Gauges [29]
- **The direction of measurement:**
  Exist strain gauge capable of measuring deformation only in one direction and the strain gauge composed of multiple overlapping elements that are able to measure forces in multiple directions. refer to Figure 3.20
- **The material on which can be pasted**

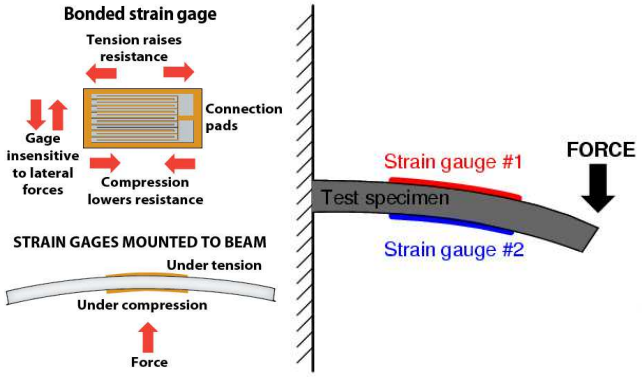
For all the torque sensors has been designed chose one Strain Gauge wire (Copper-Nickel) type general-purpose.
in particular has been used KFG-2-120-C1-23 product from KYOWA.
The sensor has the following characteristics:

- Dimension of package: 6.3X2.8mm
- Dimension the sensitive area: 2X1.2mm
- Material which can be pasted: Aluminum
- $GageFactor$ : $2.10 \pm 1.0\%$
- $GageResistance(24C, 50\%RH)$ : $120.2 \pm 0.2\Omega$
- Temperature coefficient of the gage factor:: $+0.008\%/C$

### Dimensioning of the sensors

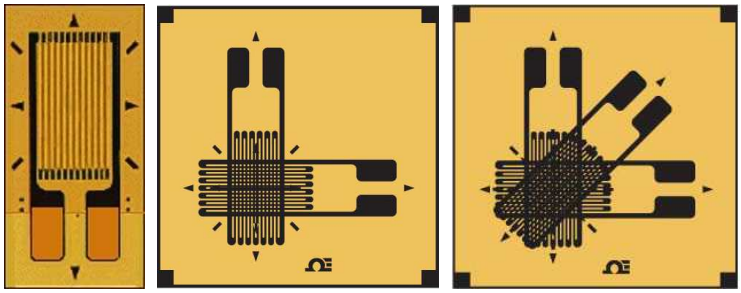
The design of all the sensors was performed using the environment "Study design" integrated in SolidWorks which provides the designer a tool optimization FEM based on the analysis of the structure to optimize. In particular, the fixed base structure of each sensor, it was possible to obtain the characteristic size (size of the strain gauge) imposing constraints on:

- Safety factor (related to the yield strength of the material)
- Minimum deformation of the area where it is glued Strain Gauge
- Maximum total deformation of the structure




(a) *Strain Gauge.* (b) *Force measurement using strain gauge.*

Figure 3.19: Operating principle based sensor Strain Gauge



(a) *Single sensor.* (b) *Double sensor.* (c) *Triple sensor.*

Figure 3.20: Strain gauge

And as imposing maximization objective function of the deformation in the vicinity of the Strain Gauge

For each variable optimization, you specify the range of variation allowed and a step. E 'can use two different methods of optimization:

- An approximate method is chosen in which a reduced number of combinations of optimization variables in their range and for each of them is performed FEM analysis. The results obtained are then interpolated to obtain an estimate of the values of the objective function and constraints for each possible combination
- A brute-force method which analyze all possible combinations of optimiza-

tion variables within the range and with the pace of change specified.

In order to optimize the process for each sensor was first used the approximate method considering a wide range of variation, then the method Brute-force in the neighborhood of the optimal point found with the approximate method and reducing the step.

### Mechanical design of torque sensors of the mobile base

One of the aspects that makes the innovative platform designed is the presence of torque sensors not only in the arm, but also in the base of the robot. In particular, in order to reconstruct the forces acting on the basis it was decided to measure the torque acting on each wheel by means of sensors mounted between the body of the motor and the base itself.
The project specifications required are as follows :

- The sensor had to be itself the engine support each wheel.
- The sensor had to be very compact.
- The sensor must be able to mechanically isolate all solicitations. In particular, it had only measure the torque acting along the axis of the engine and isolate the most of all the other forces and couples.
- Measuring range: [-2.5, 2.5] Nm with a resolution of 5 mNm.

As a result of FEM analysis of various types of geometry was chosen geometry visible in (fig 3.22). This geometry allowed to comply with all design specifications required. In particular, the deforming elements (slats on which were pasted the Strain Gauge) were produced, following the phase of study design, with a depth of 15mm and a thickness of only 0.8 mm. This ensured maximum deformation due to a torque applied along the axis of the joint and a minimum deformation due to other forces. In particular in figure 3.23 highlights the deformation due to the torque acting and a force of 20 Nm applied to the base of the sensor that is intended to simulate the weight force acting on each wheel of the robot. From the analysis it is clear that the deformation due to the force lost is about an order of magnitude smaller than that due to the torque acting along the axis of the joint. This result was also validated experimentally.

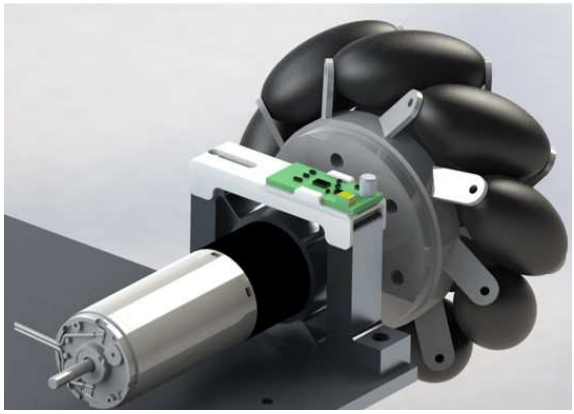


Figure 3.21: Rendering the sensor mounted

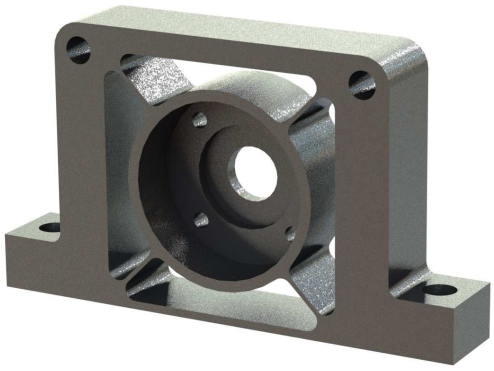
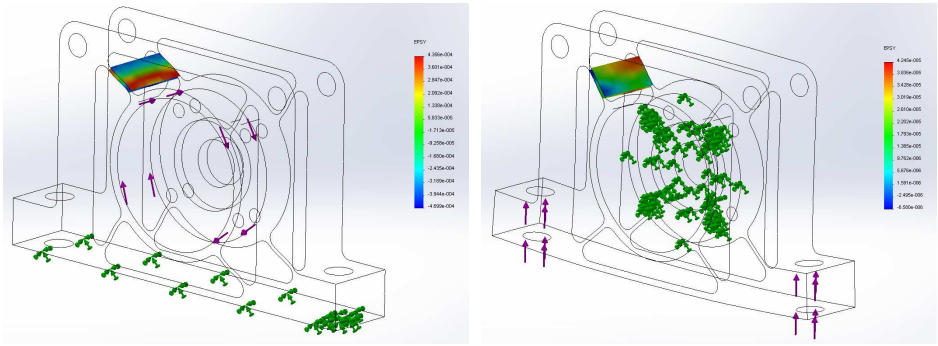


Figure 3.22: Rendering of the geometry chosen for the sensors of the base



(a) *Torque 2.5Nm.* (b) *weight force 20Nm.*

Figure 3.23: FEM analysis of the sensors based

**Mechanical design of the sensors torque of the arm**

The design specifications required are as follows:

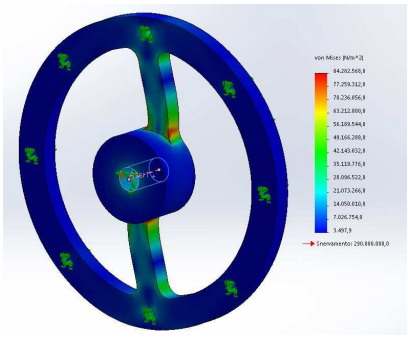
- Integration of each sensor in the geometry of the respective joint.
- It was not request a geometry able to isolate only the torque along the axis of the coupling as the isolation was guaranteed by the structure of the joint itself.
- Maximizing the deformation on the blade and the minimization of the elasticity of the sensor
- Measuring range compatible with the range of torque the motor of each joint

Choice of geometry for the sensors to the joint

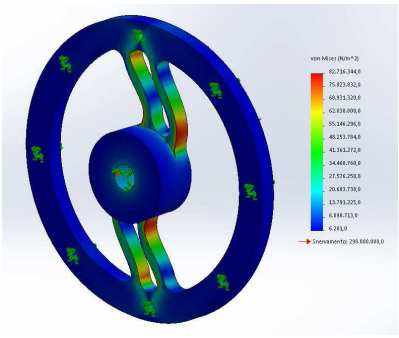
As a result of FEM analysis of various types of geometry (fig 3.24) was chosen geometry rhombus visible(fig 3.24-c). This geometry allowed to comply with all design specifications required. In particular the structure of the strain gauge like diamond is the same used in the robot LightWeight produced by Kuka and has the advantage of concentrating the deformation due to a torque along the axis of the sensor in a specific and limited area of the blade, minimizing the same time the elasticity and the stress to which the structure is subjected. From the FEM analysis carried out on the sensor model of the joint four of the robot (fig3.25 and fig3.26) shows in fact that at the same stress geometry to rumble focuses the deformation in a clearly delimited area of the blade, which of course will be glued the Strain Gauge.

(a) *First geometry.* (b) *Second geometry.* (c) *Third geometry.*

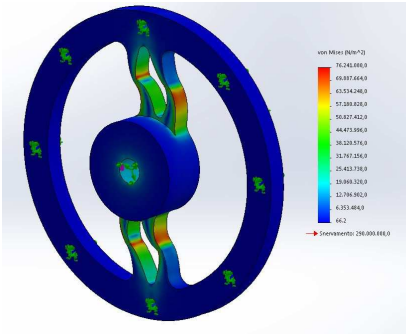
Figure 3.24: Geometries evaluated



(a) *First geometry.*

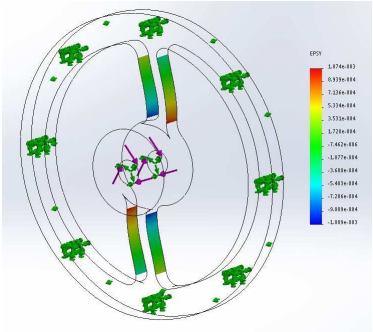


(b) *Second geometry.*

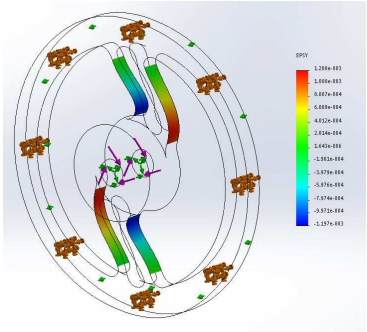


(c) *Third geometry.*

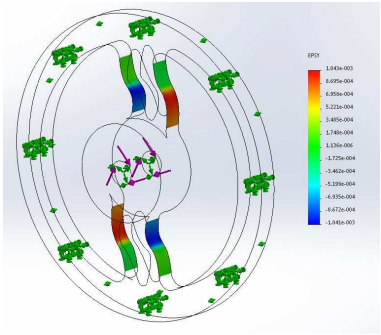
Figure 3.25: Mechanical stress



(a) *First geometry.*



(b) *Second geometry.*



(c) *Third geometry.*

Figure 3.26: Deformations

### Acquisition board

As previously said, a variation in length of the strain gauge causes a variation of resistance. In order to measure this change in resistance is designed acquisition board whose principle diagram is shown in Fig 3.27.

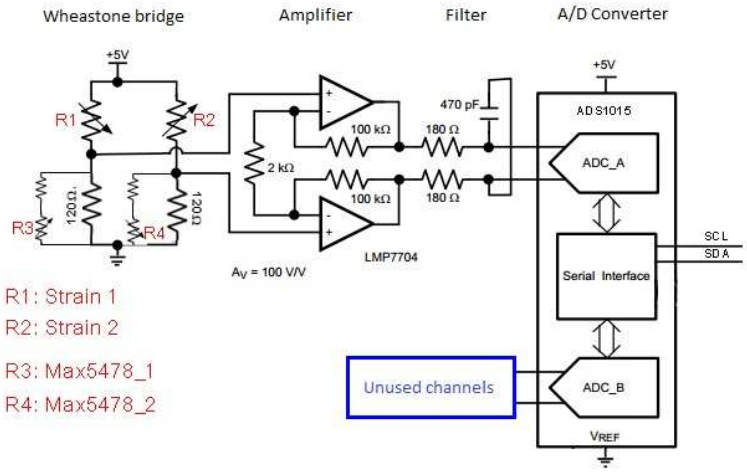



Figure 3.27: Block diagram of the conversion board

The board is composed of three stages:

- Wheastone bridge end offset regulation system
- Stage of amplification and filtering
- A/D converter

**Wheastone bridge:**

In electronics the Wheatstone bridge is an electrical-electronic equipment used to accurately measure the value of an electric resistance [34] [23]. he bridge is formed by resistors four of which in its basic version three have a known value and one is the resistor to be measured (fig 3.28). If the bridge is supplied to the voltage E output will generate a potential difference [$e_0$] given by the following relation:

$$e_0 = \frac{R_1R_3 - R_2R_4}{(R_1 + R_2)(R_3 + R_4)} * E \tag{3.4}$$

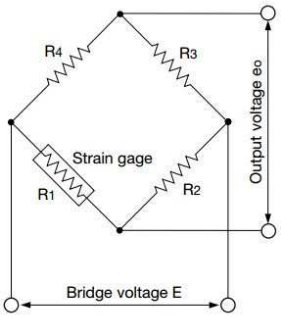



Figure 3.28: Wheastone bridge

There are various configurations of the bridge related to the number of fixed and variable resistors of the bridge and at their disposal. The following are the most important.

- Configure with a single strain gauge (fig3.29-a):
  In this configuration, only a resistance of the bridge is variable, the others are fixed and known value. The output voltage is then linked to the variation $\Delta R$ of the resistance to be measured by the following relationship:

  $$e_0 = \frac{(R_1 + \Delta R)R_3 - R_2R_4}{(R_1 + \Delta R + R_2)(R_3 + R_4)} * E \tag{3.5}$$

  If the resistors are all the same ($R_1 = R_2 = R_3 = R_4 = R$) then

  $$e_0 = \frac{R^2 + R\Delta R - R^2}{(2R + \Delta R)2R} * E \tag{3.6}$$

  and then since $R >> \Delta R$ and that $\frac{\Delta R}{R} = K_s * \epsilon$ can derive the relationship between deformation of Strain Gauge $\epsilon$ and output voltage from the bridge:

$$e_0 = \frac{1}{4} * \frac{\Delta R}{R} * E = \frac{1}{4} * K_s * \epsilon * E \tag{3.7}$$

In this configuration there is no temperature compensation.

- Configuration with dual strain gauge (fig3.29-b):
  In this configuration the two resistors in the lower part of the bridge are variable, the others are fixed and known value. The two strain gauge must be glued on opposite faces of the element deformable. The output voltage is then linked to the deformation $\epsilon$ by the following relation:

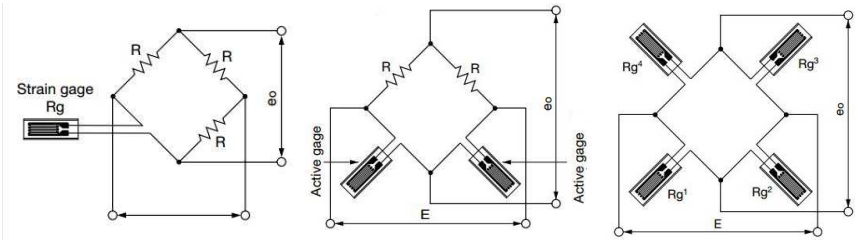
$$e_0 = \frac{1}{2} * \frac{\Delta R}{R} * E = \frac{1}{2} * K_s * \epsilon * E \tag{3.8}$$

  The resolution is double compared to the previous configuration in that the output voltage is double, also there is temperature compensation. In fact a temperature variation on the blade would vary the resistance of both strain gauge of a same amount which, as for the bridge is arranged not involve any unbalance.

- Full bridge configuration (fig3.29-c):
  In this configuration, all four resistors of the bridge are variable. The output voltage is then linked to the deformation $\epsilon$ the following relation:

$$e_0 = \frac{\Delta R}{R} * E = K_s * \epsilon * E \tag{3.9}$$

  The resolution is four times compared to the configuration with a single strain gauge, and there is temperature compensation.




(a) *One gauge.* (b) *Two gauge.* (c) *Full bridge.*

Figure 3.29: Configurazioni del ponte

**System offset adjustment:**

The offset adjustment is necessary because of the inevitable inaccuracies in the sensor causing the imbalance of the bridge even when there is no applied force. In particular:

- tolerance machining of the sensor: the blade on which is glued to the strain gauge is not perfectly flat.
- not perfectly uniform layer of glue
- positioning of the sensors are not perfectly symmetrical

In order to maximize the measurement range of the sensor has been devised a system of adjustment of the offset hardware alongside the adjustment software. Trivially in parallel to the two strain gauge in the lower part of the bridge were placed, for each one, in series with a fixed resistance by 10KΩ and a variable resistance. The two variable resistors are contained in the chip Max5478 and are controlled via i2c protocol in a range [0, 50KΩ] whit 256 steps. In this way it was possible to obtain a variation of the resistance of the two branches of the bridge amounts to 0.24Ω enough to compensate for any imbalance of the bridge itself.

**Stage of amplification and filtering:**

The output voltage of the Wheatstone bridge is less than 1mV, whereas in fact the full bridge configuration, the equation 3.9 shows that, for a deformation of the sensor approximately 0.001mm, which represents approximately the target value for each sensor designed to full scale, whereas the length of the sensitive element of the strain gauge equal to 5mm and considering that the bridge is fed to the voltage of 5V is obtained that the output voltage is approximately equal to 0.4 mV. Therefore in order to make it compatible with the measurement range of the converter A/D (3.3V) it is necessary to amplify approximately 1600X.
Therefore it has been designed a first amplification stage based on the amplifier operational LMP7704 mounted in a differential configuration with a gain equal to 100X, whereas the resistance values represented in the diagram in figure (3.27). Reality for every sensor has been found the gain value more appropriate in order to optimize the scale and resolution, by acting on the strength of feed forward. In the chapter on calibration are given the gains obtained for each sensor.
In the output of the first amplification stage was added a low pass filter with a cut-off frequency equal to:

$$f_c = \frac{1}{RC} = \frac{1}{180\Omega * 470pF} = 11.8Mhz \tag{3.10}$$

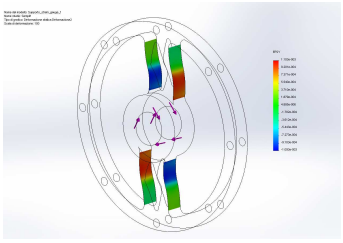
**Convertitore analogico digitale:**

The output voltage from the stage of amplification and filtering is sent to the A/D converter ADS1015 which has built a first amplifier stage with gain can be set by software to the following values [2/3X, 1X, 2X, 4X, 8X, 16X]. This choice was made in order to allow a degree of freedom software for the management of the optimal gain for each sensor. The voltage thus amplified is then sent to the A/D converter. The following are the main characteristics of the converter ADS1015

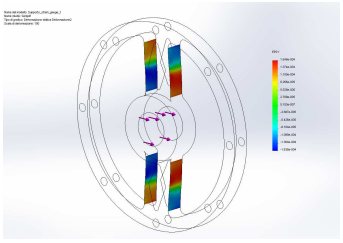
- Power supply in range 2.0-5.5 V
- Lower current assumption: 150 uA
- Sampling frequency programmable in the range from 128 to 3300 KSPS
- Four channel single-ended or two differential channel
- 12 bit
- Interface protocol I2C
- Programmable amplification stage (PGA) with possible value:[2/3X, 1X, 2X, 4X, 8X, 16X]

### Design issues: eccentricity

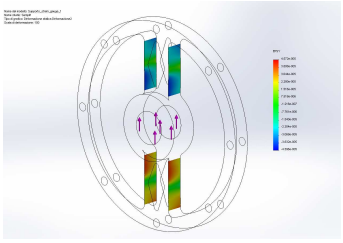
During the testing phase of joint torque sensors has been noted a problem that had not been assessed in the design phase: the non perfect concentric with the axis of the motor respect to the joint, due to machining tolerances, the forces generated on the sensor directed orthogonally to the axis of the sensor itself. This caused a Erroneous measurement depends on the angle of the joint. Below are the simulations carried out to evaluate the deformation on the gills due to these forces.



(a) *First case: Deformation due to a torque of* 4 *Nm (desired).*



(b) *Second case: Deformation due to a force of* 10 *N along the X axis (not desired)).*



(c) *Third case: Deformation due to a force of 10N along the Y axis (not desired).*

Figure 3.30: Rating problem eccentricity

To solve the problem have been used four strain gauge for each sensor to form the four branches of the bridge as shown in figure (3.32). It is observed (fig 3.31)that, from the time that the amplifier is mounted in a differential configuration, only the **Case i** leads to an actual unbalance of the bridge. In **Case ii** readings due to the efforts on the opposite faces are compensated, in **Case iii** instead you have a simultaneous movement of the two potentials of the bridge in the same direction which does not cause any reading. In fact in the **Case iii** imbalance simultaneous two potential cause a reduction of the full scale of the sensor. However, as is evident from the simulations, the deformation in the **Case iii** is approximately an order of magnitude lower than that in the other cases and therefore the reduction of the full scale due to the unbalance, even if present, is

considered negligible. Experimental tests have confirmed a reduction of the full scale of less than 1 %.

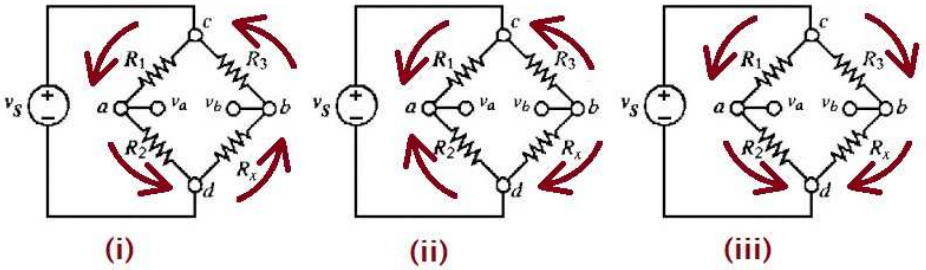



Figure 3.31: Unbalance of the bridge in the various cases

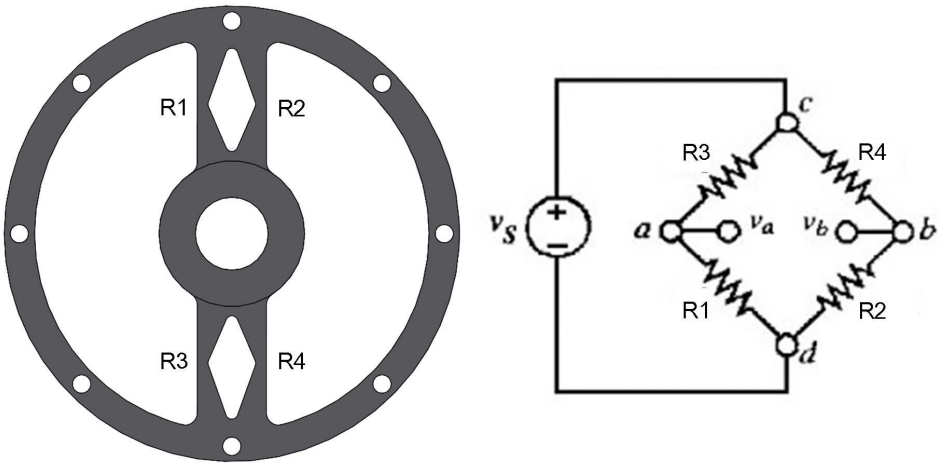



Figure 3.32: Disposizione degli strain gauge nella configurazione full-bridge

**Final realization**

Shows the photo of torque sensors complete capture system board (fig 3.33 and fig 3.34). The electronic boards of the sensors at the joints have been designed to be perfectly integrated in the geometry of each sensor in order to minimize the overall dimensions and to minimize the connection cables between the strain gauge and the acquisition card of the signals.

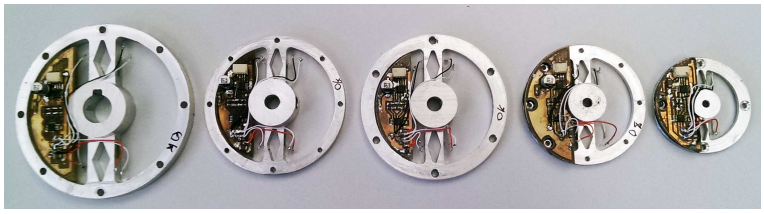


Figure 3.33: Sensors joints complete capture board

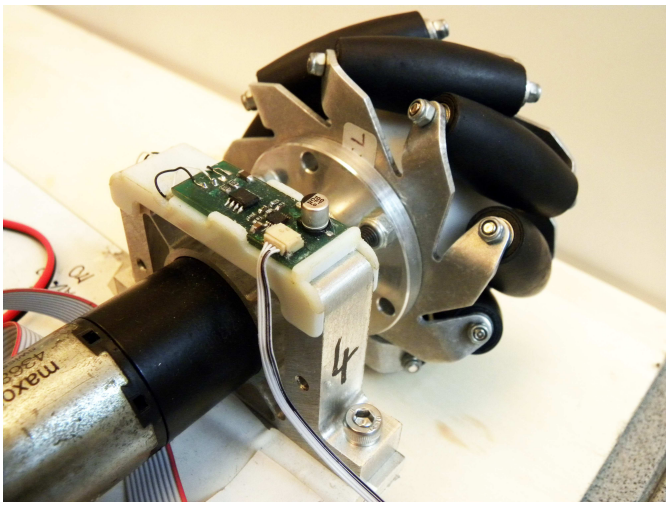


Figure 3.34: Torque sensor of the wheels of the base mounted

### 3.2.4 Platform of calibration of torque sensors

The output signal from each sensor is a voltage signal, therefore it was necessary to find the relationship between the signal and the torque measured by the sensor. To this end has been designed and built a platform calibration consists of the following components:

- Platform mechanical aluminum used to house the sensors to be calibrated
- LPC1768 mbed board used to acquire the voltage signal output from each sensor
- Sensor ATI Mini 45 used to measure the torque applied in order to find the gain with respect to the voltage signal
- Software LabVIEW used to acquire data from the mbed board and sensor ATI, display them and save them for further processing in Matlab
- Script matlab used to calculate the gain between torque and tension

The ATI sensor is placed coaxially in series with the sensor to calibrate, is applied to it a torque which is transmitted through a lever therefore also the sensor to calibrate. A system of ball bearings ensures that the force applied to the lever will result only in a pair along the axis of the sensor. In this way both sensors will be subjected to the same torque and therefore, note the size of the sensor ATI is possible to calculate the gain.

Figure 3.35: Rendering of the platform

### Progettazione e costruzione meccanica

La struttura meccanica stata progettata in modo da permettere l'alloggiamento di tutti i tipi di sensori costruiti, di qualsiasi dimensione e con qualsiasi configurazione dei fori di fissaggio. A tal fine stato progettato un sistema di asole e di perni (fig 3.36-b) tale da permetere un fissaggio semplice e veloce del sensore da calibrare alla piattaforma. La struttura stata costruita utilizzando la macchina CNC in modo da garantire la massima precisione possibile.

### Reference sensor ATI mini 45

The sensor used is the mini ATI 45 and has the following characteristics:

- Cross-type sensor compact end low profile.
- Capable of measuring forces and torques along the three axes.
- Use strain gauge silicon that require amplification considerably lower than those based on wire Copper-Nickel, this guarantees a very low noise
- he full scale values depend on the resolution and calibration performed 3.1. In our case we used the calibration SI-290-10

| Calibration | $F_x$,$F_y$ | $F_z$ | $T_x$, $T_y$ | $T_z$ |
|---|---|---|---|---|
| SI-145-5 | 145N | 290N | 5Nm | 5Nm |
| SI-145-10 | 290N | 580N | 10Nm | 10Nm |
| SI-145-20 | 580N | 1160N | 20Nm | 20Nm |

Table 3.1: Table of sensor calibration ATI (full scale)

| Calibration | $R(F_x, F_y)$ | $R(F_z)$ | $R(T_x, T_y)$ | $R(T_z)$ |
|---|---|---|---|---|
| SI-145-5 | 1/16N | 1/16N | 1/752Nm | 1/1504Nm |
| SI-145-10 | 1/8N | 1/8N | 1/376Nm | 1/752Nm |
| SI-145-20 | 1/4N | 1/4N | 1/188Nm | 1/376Nm |

Table 3.2: Table of ATI sensor calibration (resolution)

### Software labview for communication

In order to implement communication with the sensor and with the microprocessor ATI MBED, instructed to read the voltage of the sensor to be calibrated, a scheme has been developed in LabView. This solution was preferred to a solution Matlab because of the simplicity of use of such a platform for communication with heterogeneous entities. In fact communication with the sensor to calibrate occurs

via serial port using the drivers VISA while the communication with the sensor ATI is via LAN network by means of the UDP protocol. In LabView blocks are available in charge of the creation and exchange of messages for both protocols. The software sends the commands to first start the calibration offset to both sensors and then initiates a read cycle of the measured values with a frequency of 100 Hz. Finally saves data to a file for further processing in Matlab.

### Script matlab

In order to calculate the gain between the voltage and the torque for each sensor has been realized a simple matlab script. In particular, the gain was calculated using interpolation to mini square realized by functions **polyval** end **polifit** of Matlab. In addition to the total gain have been calculated, for each sensor, the gain also positive and the negative gain because there has been a slight non-linearity in some sensors that caused a small discrepancy between the two gains. The script performs the following steps in sequence:

- Import data from files that were saved by the software Labview
- Subtraction residual offset both measures
- Calculation of total gain interpolation using least squares
- Calculation of positive gain
- Calculation of negative gain
- Plot measurement before and after calibration
- Plot graph ratio voltage/torque

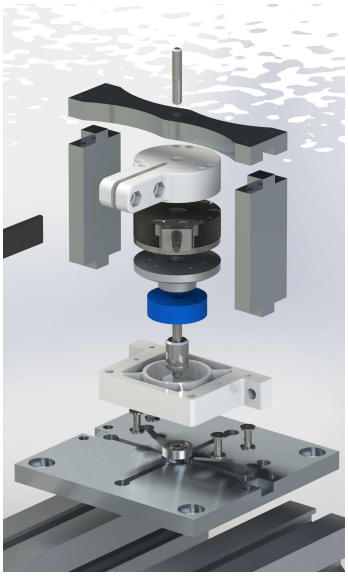
The function **p = polyfit(X,Y,n)** returns the coefficients of a polynomial p(x) of degree n which approximates better, in the sense of least squares, the curve defined by the variables x and y. In case you used a polynomial of degree 1 in order to find the best approximating straight.

The function **Yr = polyval(p,X)** returns the values of the polynomial coefficients defined by p variables x. In particular in this case will return the y components of the straight line calculated previously.

Finally, the gain K is calculated as the inverse of the angular coefficient of the straight line defined by X and Yr by means of the following relation:

$$K = \frac{1}{\frac{(Y_r(end)-Y_r(1))}{(X(end)-X(1))}} \tag{3.11}$$


(a) *Foto piattaforma durante la fase di calibrazione di uno dei sensori della base.*



(b) *Esploso piattaforma.*

Figure 3.36: Calibration platform

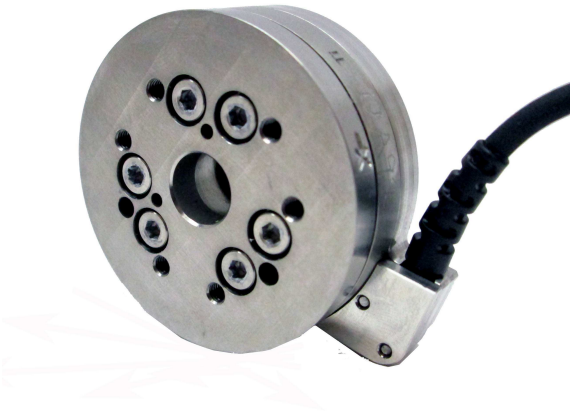


Figure 3.37: ATI mini 45

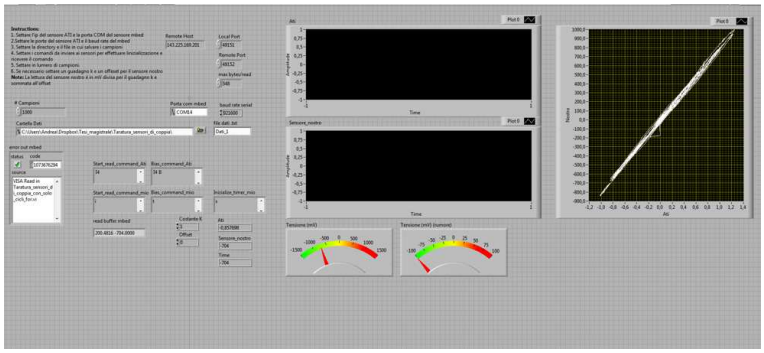
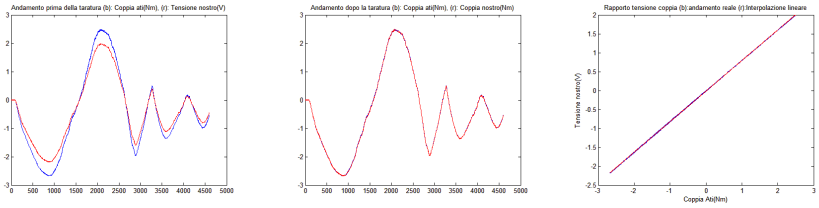


Figure 3.38: Main screen of one acquisition in Labview

### Sensor calibration

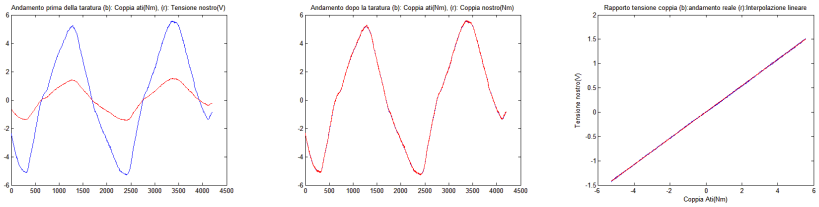
The calibration of the sensors was performed by applying on the lever connected to the sensors a variable force in the range of useful measure of each of them. The collected data were then processed in matlab to calculate the gain. Moreover, for each sensor were obtained experimentally the optimal values of the gains of the two amplifiers in the data acquisition card in order to ensure the measuring range appropriate maximizing the resolution. In particular for the first stage of amplification gain has been set by changing the resistance of the feed-back, for the amplifier contained in the A / D converter the gain is set by software. In figure 3.39 he graphs obtained matlab for one of the sensors of the wheels of the base, in particular those of the wheel 2. In figure 3.39-a) are reported trends torque sensor ATI and voltage sensor to be calibrated before calibration, in figure 3.39-b) shows again the ' torque curve of the two sensors after calibration, in figure3.39-c) shows the trend of the torque / tension and linear interpolation to the least squares obtained in order to calculate the gain.




(a) *Segnali ruota prima della taratura.* (b) *Segnali ruota dopo la taratura.* (c) *Rapporto tensione coppia e retta interpolante.*

Figure 3.39: Taratura sensori ruote

In figura3.40 invece si riportano quelli relativi ad uno dei sensori dei giunti del braccio, in particolare il giunto 1.




(a) *Segnali giunto prima della taratura.* (b) *Segnali giunto dopo la taratura.* (c) *Rapporto tensione coppia e retta interpolante.*

Figure 3.40: Sensor calibration of the joint

Can be observed as the torque response of sensors designed to be almost linear in the whole measurement range.

For each sensor were calculated:

- Total gain
- Gain positive
- Gain negative
- Span
- Resolution

The tables below show all the values calculated and also the gains of the amplifiers set.

| | $K$ | $K_{pos}$ | $K_{neg}$ | $Gain_{op}$ | $Gain_{ADC}$ | FS(Nm) | Ris(mNm) |
|---|---|---|---|---|---|---|---|
| Wheel 1 | 0.0012374 | 0.0012511 | 0.0012262 | 100 | X16 | 5.06 | 2.47 |
| Wheel 2 | 0.0009724 | 0.0009546 | 0.0009876 | 100 | X16 | 3.97 | 1.94 |
| Wheel 3 | 0.0007355 | 0.0007555 | 0.0007133 | 100 | X16 | 3.01 | 1.46 |
| Wheel 4 | 0.0007535 | 0.0007618 | 0.0007464 | 100 | X16 | 3.08 | 1.51 |

Table 3.3: Table calibration of the sensors of the mobile base

| | $K$ | $K_{pos}$ | $K_{neg}$ | $Gain_{op}$ | $Gain_{ADC}$ | FS(Nm) | Ris(mNm) |
|---|---|---|---|---|---|---|---|
| Joint 1 | 0.0010918 | 0.0010859 | 0.0010950 | 42.5 | X8 | 4.46 | 2.18 |
| Joint 2 | 0.0037017 | 0.0036985 | 0.0037161 | 100 | X8 | 15.14 | 7.39 |
| Joint 3 | 0.0019196 | 0.0019237 | 0.0019182 | 60.6 | X4 | 7.85 | 3.83 |
| Jonti 4 | 0.0005193 | 0.0005157 | 0.0005247 | 100 | X4 | 2.12 | 1.03 |
| Joint 5 | 0.0000914 | 0.0000903 | 0.0000926 | 42.5 | X8 | 0.37 | 0.18 |

Table 3.4: Table calibration of the sensors of the robot arm

## 3.3 Kinematic model of the base

From a kinematic point of view it is possible to classify mobile robots in three different categories:

- **Robot with holonomic constraints:** he configuration space of the robot is reduced, the robot can not reach any configuration in space. The constraints are presented in the following form:

$$h_i(\mathbf{q}) = \mathbf{0} \qquad i = 1, ..., k < n \tag{3.12}$$

- **Robot with anholonomic constraints:** There is no loss of accessibility in configuration space but have purely kinematic constraints. The constraints limit the mobility of the local robot preventing the assignment of a vector

of any speed in any configuration. The constraints are presented in the following form:

$$a_i(\mathbf{q}, \mathbf{q'}) = \mathbf{0} \qquad i = 1, ..., k < n \tag{3.13}$$

Or in a linear form in said form-speed *Pfaffian*

$$a_i^T(\mathbf{q})\mathbf{q'} = \mathbf{0} \qquad i = 1, ..., k < n \tag{3.14}$$

The speed allowed in each configuration are those belonging to the

$$\mathbf{N}(\mathbf{a}^T(\mathbf{q}). \tag{3.15}$$

- **Holonomic robot without constraints:** You do not have nor loss of accessibility in configuration space nor local. In any configuration, you can assign any velocity vector. These robots are said also omnidirectional.

A robot is called *Olonomo* if the DOF controllable coincide with the DOF of the robot itself.
The mobile base of the robot designed in this paper is, as previously mentioned, a base omnidirectional, therefore, at any instant of time can be assigned to any of the velocity vector:

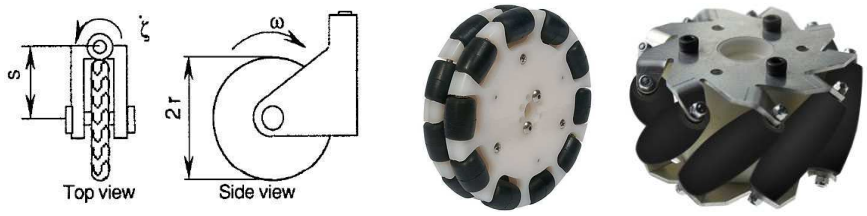
$$\mathbf{q'} = \begin{bmatrix} x' \\ y' \\ w \end{bmatrix}.$$

### 3.3.1 Holonomic wheels

There are three types of wheels that can be used to construct a holonomic robot omnidirectional:

- Wheels non holonomic controlled orientation around the vertical axis (fig 3.41-a)
- Tetrix olonome wheel (fig 3.41-b)
- Meccanum wheel (fig 3.41-c)

The control wheels anolonome is synthesized in [**?**]. The wheels Tetrix can be used in various configurations with three or four wheels. In (fig 3.42) shows the difference between a base with four wheel Tetrix or Meccanum.




(a) *Wheel nonholonomic controlled orientation.* (b) *Wheels Tetrix).* (c) *Wheels Meccanum).*

Figure 3.41: Wheels for olonome base

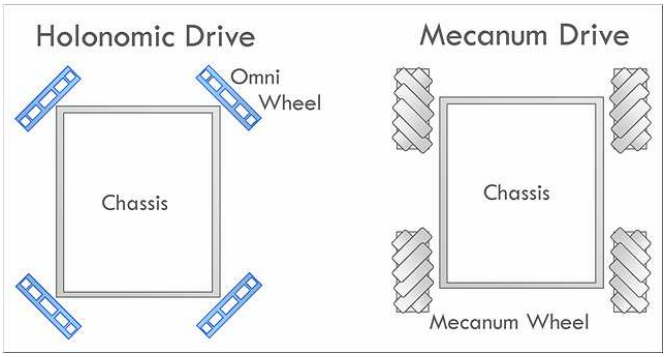



Figure 3.42: Wheels tetrix end Wheels Meccanum

The wheels of the base of the robot are designed Meccanum Wheel produced by Nexus Robot and have the rollers arranged in 45° respect to the axis of the wheel. The rotation of each wheel generates a velocity component along the x axis of the base and along the y axis, as is visible in Figure (3.44), these components allow the motion in any direction of the configuration space of the robot (fig 3.43). The base was built in such a way that the points of contact between the wheel and the floor are in correspondence with the vertices of a square of side $2l \Rightarrow L = l$.

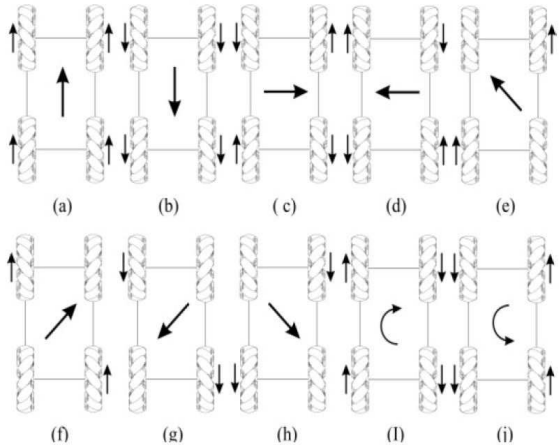



Figure 3.43: Motion of a wheeled base meccanum

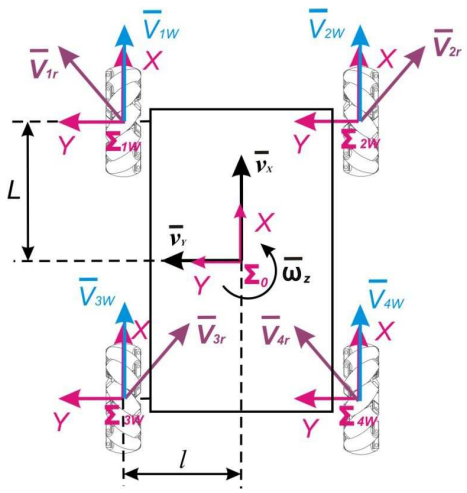



Figure 3.44: The velocity components of a base with wheels meccanum

The kinematic model was calculated by reference to [**?**],[**?**] end [**?**] in which describes the behavior of the wheels Meccanum-wheel and the kinematic and dynamic model of an omnidirectional base.
Referring to Figure 3.45 the following table shows the transformation matrix and the Jacobian matrix.

$$T_b^o = \begin{bmatrix} cos(\alpha) & -sin(\alpha) & 0 & d1 \\ sin(\alpha) & cos(\alpha) & 0 & d2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$J = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Where $d_1, d_2, \alpha$ respectively represent translations along the axes x and y of the triad base and the rotation about the z axis

The relationship between the variables of the joint and the positions of the wheels does not exist in direct form but exists only in differential form between their speed. This report is given by the Jacobian matrix $S$ below:

$$S = \frac{1}{r} \begin{bmatrix} -1 & -1 & 2l \\ -1 & 1 & 2l \\ 1 & -1 & 2l \\ 1 & 1 & 2l \end{bmatrix}. \tag{3.16}$$

Where $r$ is the radius of the wheel and $l$ is the seeds distance between the points of contact of the wheel with the floor. Therefore notes velocities in Cartesian space (three-base), you can calculate the speed to be assigned to the wheels of the robot projecting first on tern integral with the robot through the rotation matrix $R$ defined so:

$$R = \begin{bmatrix} cos(\alpha) & -sin(\alpha) & 0 \\ sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}^T. \tag{3.17}$$

and then projecting it on the wheels by the matrix $S$.
Finally we define the velocity vector in the basis reference

$$q' = \begin{bmatrix} \dot{d1} \\ \dot{d2} \\ \dot{\alpha} \end{bmatrix}.$$

and the velocity vector of the wheels

$$q'_r = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}.$$

then the relationship between the speed and the speed triad based on the wheels is the following:

$$q'_r = S * R * q' \tag{3.18}$$

Finally, the path to be assigned to the wheels of the robot will be found by integrating over time the speeds obtained by the previous report.

## 3.4 Experimentation

### 3.4.1 Control architecture

In (figure 3.46) is shows the structure of the skeleton of the platform management hardware. Each joint of the robot (the wheel, the joint of the arm and gripper) is controlled by a dedicated node (figure 3.47) which deals with the management of the sensors and the actuator and the low-level control. An on-board computer communicates with the various nodes through the USB bus, manages the external sensors (Kinekt, laser scanner, etc.) and the high-level control

### 3.4.2 Development of low-level control

Will be described the control loop implemented and the main classes used for the management of each node of the robot. In particular, we will focus on the classes for the management of the engine and for the management of low-level sensors.

#### Control Loop

On each node of the robot was implemented a control algorithm whose diagram is shown in Figure (3.48). The algorithm is based on a triple loop control in torque, speed and position. The first and second exploit a PI controller (Proportional-Integral), the third a simple proportional controller. In its first version of the loop pair is closed taking account of an indirect measure of the torque obtained through the measurement of current. In subsequent versions would also be the measure obtained through torque sensors by adding an 'additional loop control.
The advantage of this solution to triple loop lies in the fact of being able to control the individual joints is in position both in speed and in a couple simply opening or closing the various loops. You can also assign a feed-forward in both speed torque to improve overall performance.

### Classes for the management of the motors

Each node of the robot operates a motor. We use the following classes:

- **Class Motor:** allows the instantiation of an "object" and engine speed management control using PWM. Input to the manufacturer must supply the following:
  - Pin of PWM signal.
  - Pin of motor phase, is high if the engine has to turn clockwise otherwise low.
  - Frequency of PWM. set to 20Khz for all the engines, the maximum of the power driver.
- **Class PI2**: Handles the low-level controllers through a control PI *(Proportional-Integral)* pattern with desaturation of the integral and dynamic saturation. In particular, the saturation of the current is achieved by using a function time variant two step (3.19) which takes into account the estimated temperature of the motor so as to allow current peaks higher than the rated current for short periods of time.

$$I_s = SF*I_{max}*(2.5+tanh(dt-T_{sat-Imax})*(dt > T_{sat-Imax})+(dt-SF*T_{sat-Imax})*(dt > SF*T_{sat-Imax}) \tag{3.19}$$

Where:

- $SF$: Scale factor
- $T_{sat-Imax}$: Period
- $dt$: Time elapsed from the instant when the current exceeds the threshold $I_s$
- $Imax$: Limit current of the motor

The measured current $I$ is compared with the saturation current $I_s$, if the is generated exceeds a feedback signal voltage (with gain $G_{si}$) such as to reduce the motor voltage and thus the current absorbed , according to the following law:

$$\left\{ \begin{array}{cc} V_i = G_{si}(I - I_s) & if I > I_s \\ [\ V_i = G_{si}(I + I_s) & if I < -I_s \\ V_i = 0 & otherwise \end{array} \right] \tag{3.20}$$

### Management of sensors

Each node of the robot operates three sensors:

- Current sensor
- Encoder
- Torque sensor

**Current measure :**

The current measurement is performed by means of the AD8206 chip that provides an output voltage proportional to the current measured according to the following law:

$$C = \frac{T * V_{cc}}{Gain * R_c} \qquad C = Corrente, \quad T = Tensione - misurata \quad R_c = Res - feedback \tag{3.21}$$

The output voltage is therefore read on one of the pins A/D of the micro controller through the function: $AnalogIn.read()$.

**Position and speed measure:**

The position measurement is performed by the encoder is mounted on each engine. A dedicated chip is responsible for reading the encoder and Telecommunication of the angular position through the SPI protocol. The class *LS7366* is responsible for management of the chip and provides the function: *encoder.Read-CNTR()* to read the number of pulses measured. The reading is carried out in X4 mode, thus using both the rising edges that drop to increase the resolution. The angular position was finally calculated using the following relationship:

$$P = CNTR * \frac{2\pi}{ETP * 4} \qquad P = Posizione \quad CNTR = Impulsi \quad ETP = Encoder - Turn - Pulse \tag{3.22}$$

The speed measurement is obtained on the basis of the position measurement by performing an operation of derivation "dirty" and filtering

$$V = (P - P_{hold})/dt_v \qquad Derivata \tag{3.23}$$

$$V = af * V_{hold} + bf * V \qquad Filtraggio \tag{3.24}$$

**Torque measure:**

The torque measurement is performed using the torque sensors designed in the present work.
In this regard was implemented a dedicated class called *Strain* that contains all the necessary functions for reading and sensor calibration. The class *Strain* include a the *ADS1015* class for the manage of the A/D converter end the *MAX5478* class for the management of the variable resistance.
The functions of the class *Strain* are:

- **Strain.read()**: allows the reading of the torque value read by the sensor. The measurement is directly a torque as the torque-voltage conversion is carried out within the class.

- **Strain.SetOffset():** Performs the setting of the offset of the sensor by adjusting the two resistors in parallel to the bridge. In particular is exploited an algorithm described by the following steps:
  - Ascent slow of the firs resistance to the maximum allowed value and slow down of the second resistance to a minimum.
  - Ascent of the second resistance holding fixed the value of first. If the offset is less than a certain tolerance $\Longrightarrow$ **break**
  - If the first round does not find the optimum point $\Longrightarrow$ Ascent slow the second resistance to the maximum allowed value.
  - Descent of the resistance one holding fixed the value of second resistor. If the offset is below a certain tolerance. $\Longrightarrow$ **break**
  - If the good was found the values of the two resistors are saved in the register non volatile resistance otherwise it returns the error code. **-1**

  The slow steps of ascent and descent are used to prevent current peaks due to a step change of the resistance.

### 3.4.3 First test on the mobile platform

The control structure is, at present, being implemented. E 'was developed the architecture of low-level control of the mobile base (four nodes of the wheels), while the low-level architecture of the arm and the high level architecture will be implemented to follow.
To evaluate the overall performance of the mobile base (low-level control and torque sensors) has implemented a simple algorithm decentralized control. In particular, on each wheel of the movable base has been implemented a simple position control as described in Figure (3.48). The position reference $P_D$ was linked to the torque measured by the sensor $C$ by the relation (3.25) so as to implement a virtual spring, without damping, with elastic constant $K_m$.

$$P_D = \frac{1}{K_m} C \tag{3.25}$$

Finally acquired, through a Labview software and Matlab, the torque from the four sensors and has been used to the report (**??**) in order to derive the generalized forces acting on the base.

#### Results

At the mobile base were applied forces, as shown in figure (3.49), in order to simulate the generalized forces acting on the same. The spring constant has been chosen equal to $K_m = 0.125N/m$. In figure (3.50) shows the trends of the forces derived from the measurement of torque obtained from the sensors. From the graph it is possible to check how the forces acting in a certain direction are actually measured through the measurement of torque at the wheels. Obviously contributions

coupling visible are due to inability to practically apply the forces only in desired directions. The undershoot visible in the graphs is due to the elastic return of the base that since there is damping part that due to friction, has generated an oscillatory.

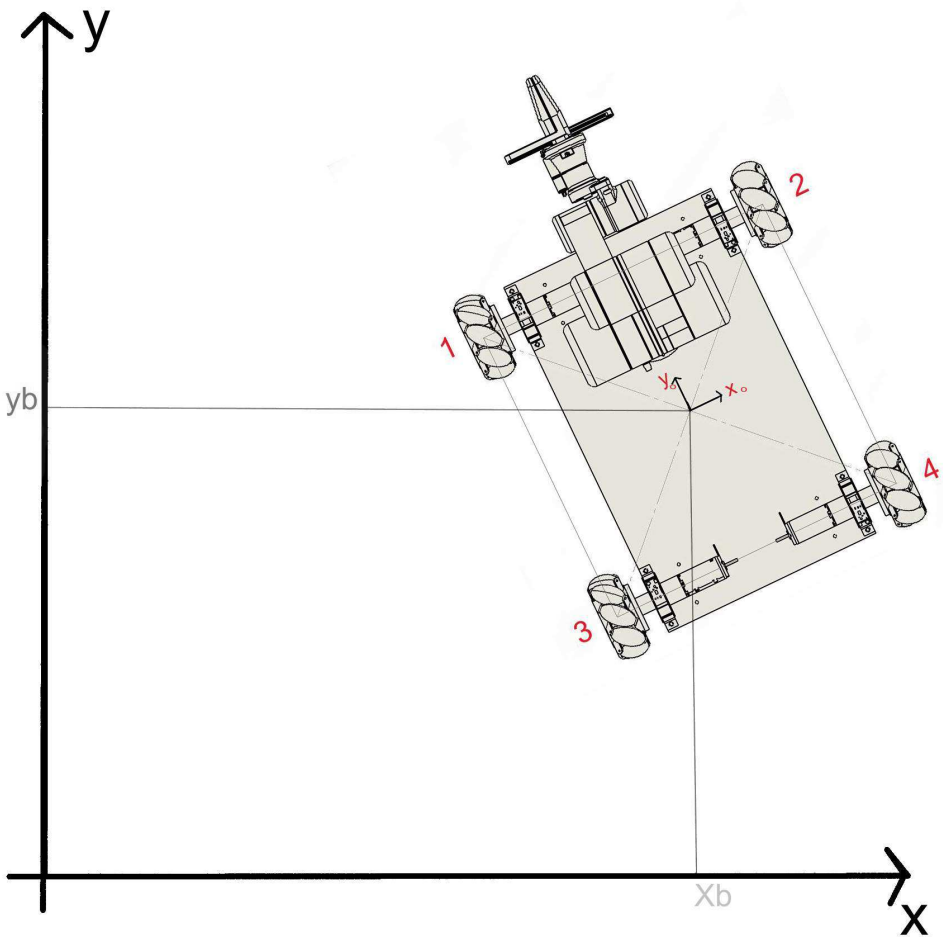



Figure 3.45: References mobile base

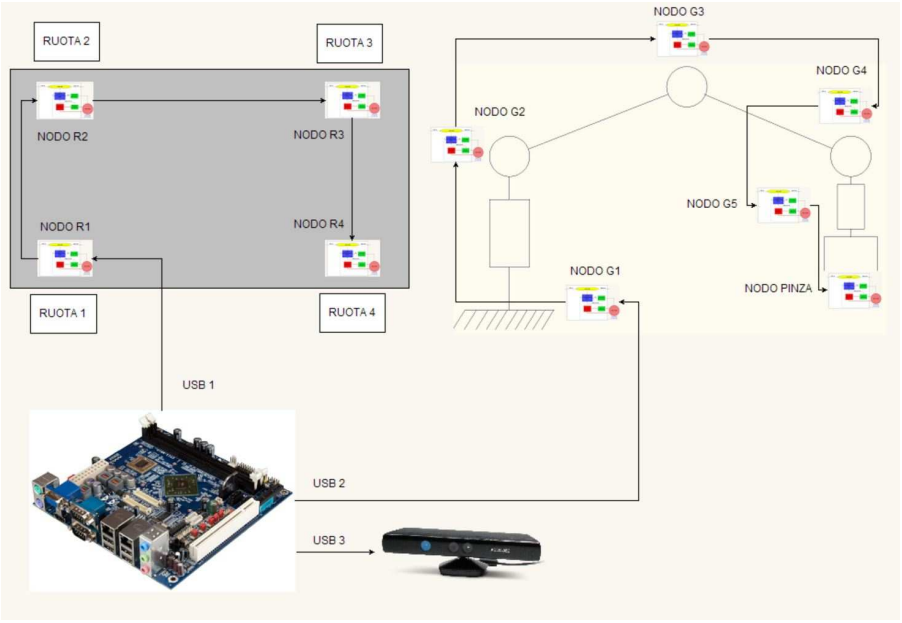



Figure 3.46: Scheletro hardware di gestione della piattaforma

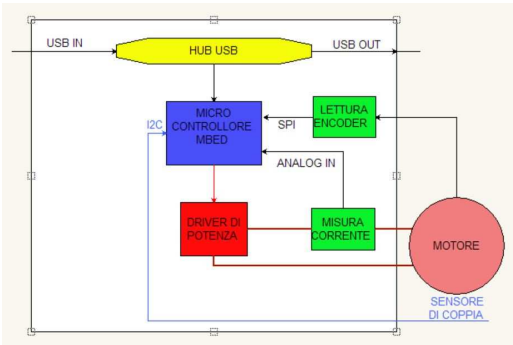



Figure 3.47: Schema di ogni nodo del robot

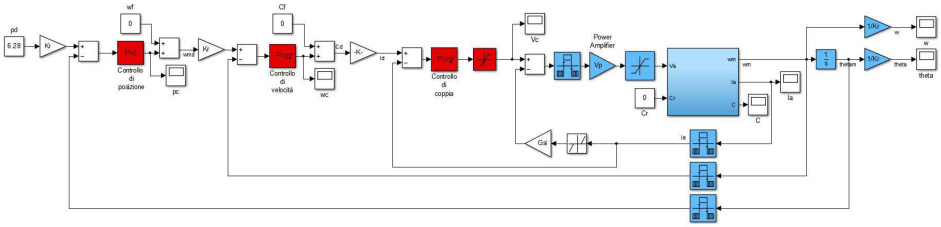


Figure 3.48: low-level control loop

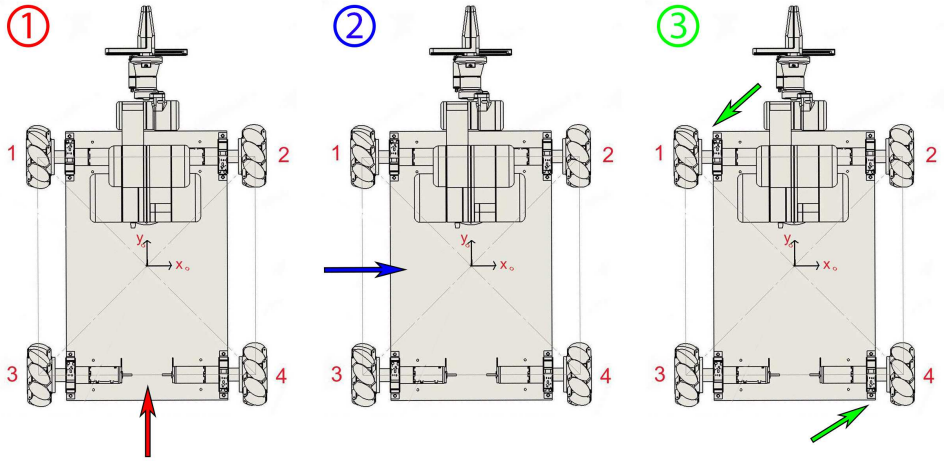



Figure 3.49: Forces applied on the mobile base

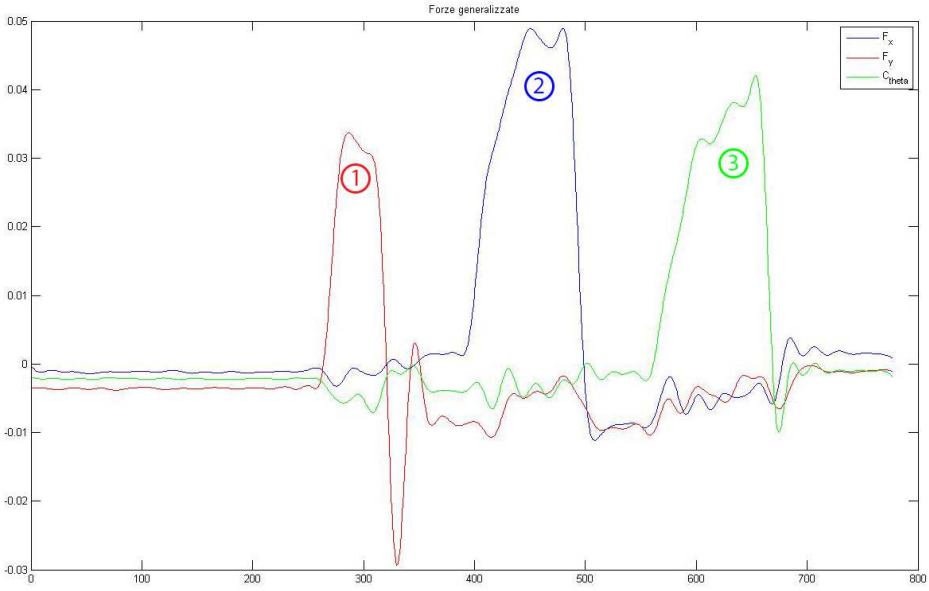



Figure 3.50: Low-level control loop

# Chapter 4

# Effects of Packet Losses on Formation Control

This work was supported by the Network for Excellence of research programme MASTRI funded by POR Campania FSE 2007-2013. The research leading to these results has been also supported by the ARCAS and SHERPA collaborative projects, which both have received funding from the European Community Seventh Framework Programme (FP7/2007-2013) under grant agreements ICT-287617 and ICT-600958, respectively.

The impact of the main packet loss models on the Unmanned Aerial Vehicle (UAV) formation control has been evaluated. A simulation environment has been built to introduce a centralized architecture, usually employed in mobile robotics to pursue global tasks, in the presence of loss models affecting the communication through robots hops. Simulation results show that the control performance in terms of tracking, collision avoidance and loss of connectivity are effected by the specific characteristic of packet loss. This suggests that the design of UAV formation control over wireless network has to be carried out strongly taking into account the effects of specific packet loss characteristics (due to wireless protocol, disturbance, traffic) on the performance quality.

## 4.1 Introduction

Development of distributed communication, computing, and control functionalities are providing the ability of monitoring and controlling complex or distributed processes by a large number of sensors, actuators, and computational units interconnected by wireless communication. These emerging network application paradigms such surveillance networks, formation flight, clusters of satellites, automated highway systems have led to the requirement of designing distributed consensus algorithms over network for estimation, detection, optimization and

control. One common feature of these researches is the sharing of information between agents in order to address a common objective. Algorithms solving the distributed robot coordination problem provide the means by which networks of agents can be coordinated. As VLSI (Very Large Scale Integration) technology advances and computing power grows, robots are becoming more and more intelligent, robust and power-efficient so that they possess the capabilities of communication and cooperative work [10]. This allows to implement wireless networked robotic system for reducing the need of human presence in dangerous applications (i.e. fire fighting, military or civilian search and rescue missions, security, surveillance). A large body of research, from various perspectives, has been produced both by ad hoc networking [8, 14] and by the robotics research communities to achieve self-organization and coordination of groups of robots through the use of optimization techniques, i.e., the one proposed by [13], consensus and rendezvous algorithms by [36, 35].

Several architectures have been exploited in robotics to solve the problem about coordinate several agents (or nodes) to accomplish a certain goal. The outputs of single behaviors are combined in the centralized Null-Space Based (NSB) approach proposed by [6, 5] to compose a complex mission for swarms of small ground mobile robots. Such approach takes inspiration from the singularity-robust task-priority closed-loop inverse kinematic (CLIK) algorithm for industrial robots [49], which has been also employed by [33, 11, 12] in many fields like robotic manipulation. On the other hand, [38] present a leader-follower formation control based on uncalibrated omnidirectional cameras. A control strategy for distributed monitoring tasks trough the so-called geometric moments of the swarm is introduced by [39]. Finally, [43] gives an overview about multiple robot systems.

For all the above reasons, it might be of great interest:

- i) introduce a simulator to evaluate well known centralized architectures usually used in mobile robotics to pursue global tasks.
- ii) take into account the limitation of the wireless communication supporting the robot cooperation.

Specifically, the presence of different packet loss models affecting the communication through the robots hops are considered and their effects on wireless networked robotic performance are evaluated. It will pointed out that in networked robotic architecture design the effect of packet loss on the overall performance has to be taken into account. Few works in the literature carry out a similar analysis. For instance, [19] analyse the impact of medium access protocol on average consensus problem over wireless networks for a group of quad-rotors. That work provides a simulation environment modelling different network communication layers, analysing the impact of the Carrier Sense Multiple Access algorithm with Collision Avoidance (CSMA/CA) and Time Division Multiple Access al-

gorithm(TDMA). As a result, the average consensus using CSMA/CA protocol presents a large convergence time due to both packets drop and time delay in the end-to-end transmission among quad-rotors, while TDMA is more suitable for real-time communication.

Otherwise from [19] that focuses on the effect of channel access algorithm on the swarm performance, herein the effect on the swarm performance evaluation of the main packet loss models presented in the literature [24] are considered. Additionally, further performance metrics (i.e. Maximum Agent Distance, Collision Avoidance, and so on) are taken into account. In such a way it is possible to highlight the problem of loss of communication between agents moving inside structured environments which is essential in aerial robotic applications. Using these models allows to analytically investigate and make more robust control algorithms. This analysis will lead to the choice of political communication in terms of traffic and network topology so as to improve the control performance.

In the section 4.2 introduces the employed centralized architecture for the the network communication and task control. In Section 4.3, the considered loss models are revised. Section 4.4 provides a performance evaluation to understand the limitation of wireless communication in a centralized architecture.

## 4.2 Modeling of the centralized architecture

### 4.2.1 Layout of the architecture

The proposed architecture is made up of $n$ agents, each of one representing the single UAV of the swarm. Without loss of generality, the $i$th vehicle is able to measure its position $\boldsymbol{p}_i \in \mathbb{R}^3$ with respect to a fixed inertia frame $\Sigma_w$, where $i = 1, \ldots, n$. Moreover, each vehicle is able to detect the position $\boldsymbol{o}_{i,k} \in \mathbb{R}^3$ of obstacles which are present within a distance $\rho_i$ from $\boldsymbol{p}_i$, where $k = 1, \ldots, o_i$ with $o_i$, the total number of obstacles detected by the vehicle $i$. An obstacle can be both another vehicle of the swarm and a generic object in the environment that should be avoided.

Considering the dynamic model of a UAV as described by [40], it is possible to recognize that the position $\boldsymbol{p}_i$ and the so called *yaw* rotation around its vertical axis are flat outputs of the system [17]. Therefore, without loss of generality, both the yaw rotation and the low-level control implemented by [40] might be neglected assuming that each UAV is controlled through its linear velocity.

Finally, a central node elaborates the measurements given by each vehicle with respect to a certain number of tasks and constraints to be fulfilled, providing online the desired trajectory that each UAV has to follow. More details are provided below.

### 4.2.2 NSB approach to control UAVs swarms

The goal of the NSB approach is to combine the outputs of elementary behaviors so as to command each robot, i.e., UAV, of the swarm. A single behavior can be defined either in a global or local fashion. If the number of degrees of freedom of the system are greater than the ones required to fulfill the main behavior, other behaviors can be considered and arranged with different priorities, leading to a hierarchical framework. The approach here presented modifies what described by [6, 5] along the lines of the algorithm introduced by [12].

In general, a behavior can be represented by either a function aiming to reach a desired value, i.e., the centroid of the swarm has to reach a fixed point in the space, or a cost function that should be maximized(minimized), i.e., maximize the distance of the UAVs from the obstacles. The former behaviors are then called *tasks*, the latter instead *constraints*.

Let $\boldsymbol{p} = \begin{bmatrix}\boldsymbol{p}_1^{\mathrm{T}} & \cdots & \boldsymbol{p}_n^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{3n}$ be the configuration of the system. A generic task can be represented trough the $j$th task variable $\boldsymbol{\sigma}_j \in \mathbb{R}^{m_j}$ to be controlled, where $j = 1, \ldots, n_t$ with $n_t$ the total number of the tasks and $m_j$ the dimension of the $j$th task variable. $\boldsymbol{\sigma}_j$ is related to the configuration of the system as follows

$$\boldsymbol{\sigma}_j = \boldsymbol{f}_j(\boldsymbol{p}), \tag{4.1}$$

with $\boldsymbol{f}_j \in \mathbb{R}^{m_j}$ a proper vector function. Notice that $j$ denotes also the priority of the task, i.e., 1 the highest priority, $n_t$ the lowest. Taking the time derivative of (4.1) yields

$$\dot{\boldsymbol{\sigma}}_j = \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{\sigma}_j}\dot{\boldsymbol{p}} = \boldsymbol{J}_j(\boldsymbol{p})\dot{\boldsymbol{p}}, \tag{4.2}$$

where $\boldsymbol{J}_j(\boldsymbol{p}) \in \mathbb{R}^{m_j \times 3n}$ is the configuration-dependent task Jacobian matrix.

On the other hand, each constraint can be described through a cost function $\mathcal{C}_h(\boldsymbol{p})$, where $h = 1, \ldots, n_c$, with $n_c$ the total number of constraints. Each function $\mathcal{C}_h$ can increase its value when the configuration of the system is near to violate the constraint. In order to minimize such cost function, the swarm can be moved accordingly to the descend gradient $-\nabla_{\boldsymbol{p}}^{\mathrm{T}}\mathcal{C}_h(\boldsymbol{p})$ representing a fictitious force moving away the UAVs from the configurations violating the constraints. An overall cost function can be given by

$$\mathcal{C} = \sum_{h=1}^{n_c} \gamma_h \mathcal{C}_h(\boldsymbol{p}),$$

where $\gamma_h$ is a positive weight related to the $h$th constraint.

The following velocity control input can be chosen

$$\begin{aligned}\dot{\boldsymbol{p}} =& \boldsymbol{J}_1^{\dagger}(\boldsymbol{p})\left(\dot{\boldsymbol{\sigma}}_{1,d} + \boldsymbol{K}_1\boldsymbol{e}_{\sigma,1}\right) + \sum_{j=2}^{n_t} \boldsymbol{P}(\boldsymbol{J}_j^A(\boldsymbol{p}))\boldsymbol{J}_j^{\dagger}\boldsymbol{K}_j\boldsymbol{e}_{\sigma,j} \\ & - k_{\nabla}\boldsymbol{P}(\boldsymbol{J}_{n_t+1}^A(\boldsymbol{p}))\nabla_{\boldsymbol{p}}^{\mathrm{T}}\mathcal{C},\end{aligned} \tag{4.3}$$

where † denotes the general pseudo-inverse of a matrix, $\boldsymbol{\sigma}_{j,d}(t)$ and $\dot{\boldsymbol{\sigma}}_{j,d}(t)$ represent the desired trajectory for the $j$th task, $\boldsymbol{e}_{\sigma,j} = \boldsymbol{\sigma}_{j,d} - \boldsymbol{\sigma}_j$ is the task error vector, $\boldsymbol{K}_j \in \mathbb{R}^{m_j \times m_j}$ is a suitable positive definite gain matrix, $k_\nabla$ is a positive gain, $\boldsymbol{P}(\boldsymbol{J}_j^A(\boldsymbol{p}))$ is the matrix allowing the projection of a vector onto the null space of $\boldsymbol{J}_j^A(\boldsymbol{p})$ which is the augmented Jacobian given by

$$\boldsymbol{J}_j^A(\boldsymbol{p}) = \begin{bmatrix} \boldsymbol{J}_1^{\mathrm{T}}(\boldsymbol{p}) & \boldsymbol{J}_2^{\mathrm{T}}(\boldsymbol{p}) & \cdots & \boldsymbol{J}_{j-1}^{\mathrm{T}}(\boldsymbol{p}) \end{bmatrix}^{\mathrm{T}},$$

and which benefits of some robustness properties in the stability proof as highlighted by [4]. In order to obtain the desired motion reference $\boldsymbol{p}_d(t) = \begin{bmatrix} \boldsymbol{p}_{d,1}^{\mathrm{T}} & \cdots & \boldsymbol{p}_{d,n}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ for the UAVs, the control law in (4.3) can be integrated.

The geometric interpretation of (4.3) is that each behavior is first evaluated alone while, before adding its contribution to the overall vehicle velocity, a lower priority task is projected onto the null space of the previous task in the priority stack, in such a way as to avoid conflicts between it. The NSB approach always accomplishes the main task, while the fulfilment of the others depends on the specific situation and should be thus discussed case by case.

If the system is close to violate one or more constraints, a high-level supervisor has to remove some tasks so as to fulfil the constraints. The adopted management of removal/insertion of tasks is described by [12].

The two tasks defined in this paper are listed below in a decreasing priority order. Many others can be considered [5].

- **Swarm centroid.** The centroid of the swarm is given by the mean value of the position of each UAV

$$\boldsymbol{\sigma}_1 = \frac{1}{n} \sum_i^n \boldsymbol{p}_i, \tag{4.4}$$

  and it is controlled to track a desired trajectory $\boldsymbol{\sigma}_{1,d}(t)$ and $\dot{\boldsymbol{\sigma}}_{1,d}(t)$.

- **Formation control.** Let $\Sigma_b$ be a frame placed at the swarm's center of mass and moving with the platoon. It is possible to associate a desired position $\boldsymbol{p}_{i,d}^b(t)$to each UAV of the swarm, expressed with respect to $\Sigma_b$. The task variable is thus defined as follows

$$\boldsymbol{\sigma}_2 = \begin{bmatrix} \boldsymbol{p}_1^{b^{\mathrm{T}}} & \cdots & \boldsymbol{p}_n^{b^{\mathrm{T}}} \end{bmatrix}^{\mathrm{T}}, \tag{4.5}$$

  and it is controlled to track a desired trajectory $\boldsymbol{\sigma}_{2,d}(t)$.

The following constraint is considered in this paper.

- **Obstacle avoidance.** In order to avoid collisions between two UAVs or a vehicle and an object in the environment, it is imposed that the distance

between a UAV and an obstacle should be greater than a safety value $\rho_s$. The next cost function can be hence considered

$$\mathcal{C}(\boldsymbol{p}) = \sum_{i=1}^{n} \sum_{k=1}^{o_i} c_{i,k}(\boldsymbol{p}_i), \tag{4.6}$$

$$c_{i,k}(\boldsymbol{p}_i) = \begin{cases} k_{i,k} \frac{\rho_s - d_{i,k}(\boldsymbol{p}_i)}{d_{i,k}(\boldsymbol{p}_i)^2} & \text{if } d_{i,k}(\boldsymbol{p}_i) \leq \rho_s, \\ 0 & \text{if } d_{i,k}(\boldsymbol{p}_i) > \rho_s, \end{cases} \tag{4.7}$$

$$d_{i,k}(\boldsymbol{p}_i) = \|\boldsymbol{p}_i - \boldsymbol{o}_{i,k}\|, \tag{4.8}$$

where $k_{i,k}$ is a positive gain.

### 4.2.3 Network configuration

The proposed configuration is centralized, hence it is possible to recognize a master node and several slaves. Without loss of generality, it will be considered that the master is one of the $n$ agents. The remaining $n-1$ agents are referred to as salves. The master computes the references for each salve as in (4.3) and communicates to them trough wireless channels.

Taking into account what considered in Section 4.2.1, each UAV agent can be modelled as the following second order system

$$\dot{\boldsymbol{s}}_i = \begin{bmatrix} \dot{\boldsymbol{x}}_{1,i} \\ \dot{\boldsymbol{x}}_{2,i} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{p}}_i \\ -\omega_{n,i}^2 \boldsymbol{p}_i - 2\zeta_i \omega_{n,i} \dot{\boldsymbol{p}}_i + \omega_{n,i}^2 \boldsymbol{p}_{d,i} \end{bmatrix}, \tag{4.9}$$

with $i = 1, \ldots, n$ and where $\omega_{n,i}$ and $\zeta_i$ are the natural frequencies and damping factor for each agent.

In order to evidence the interconnections between the master and the slaves, let assume without loss of generality that the first agent is the master, while the $k$-the agent, with $k = 2, \ldots, n$, is a slave. Considering with $a_{ij} = \{1, 0\}$, with $i = 1, \ldots, n$ and $j = 1, \ldots, n$, the communication channel between agents $i$ and $j$, which could be active $a_{ij} = 1$ or not $a_{ij} = 0$, it is possible to recognize that

$$a_{ij} = \begin{cases} 1 & i = 1, j = 2, \ldots, n \\ 0 & \text{otherwise.} \end{cases}$$

## 4.3 Loss modelling

Experimental packet traces reveal that packet losses are bursty in wireless networks. In a loss burst most of the transmitted packets will be lost, while in a loss-free burst there are few or no packet losses. It is straightforward to use a two-state Markov chain to model the behavior of a wireless channel.

In the following, the main packet loss models will be introduced and considered in the performance evaluation. For more details, please refer to the related references [21, 51, 24].

### 4.3.1 Bernulli model

The simplest loss pattern is the Bernoulli model. Such model is constituted by a single loss parameter and each transmission is considered independent from the others. The loss probability is a parameter representing the measure of goodness about the transmission channel. Due to the independence of individual packages, this model can not take into account the channel memory and hence impulse noise. In the practice, the model can act in two different conditions, namely, *active link* with probability $p$ and *broken link* with probability $1 - p$.

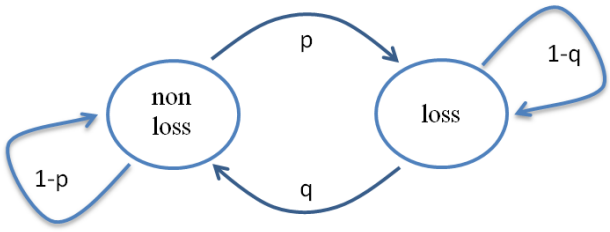



Figure 4.1: Gilbert model.

### 4.3.2 Gilbert model

Gilbert model (1-st Markov chain model) is frequently employed in the study of packet-loss process in communication networks. In such a model each channel maintains a status flag which can be either labelled as *good* or *bad*. In the former state there is very few packets loss, while in the latter most packets are lost. Gilbert model has several parameters: temporal error correlation, probabilities of a good or bad channel, probabilities of error given that the channel is in the good or the bad state. In order to mitigate the computational complexity, a simplified version of the Gilbert-Elliot model is also employed. In this version, if a packet loss occurs while the channel is in the good state, the channel immediately switches to the bad state. Similarly, whenever a packet is received successfully while the system is in the bad state, the channel switches to the good state. Let assume the probabilities of staying in the good state and the bad state are the same proposed by [24]. On the one hand, if the current packet has been successfully received, the probability that the next incoming packet is lost is denoted by $p$. On the other hand, if the current packet has been lost, the probability that the current packet is successfully received is denoted by $q$. Therefore, the probabilities $1-p$ and $1-q$ are the conditional probabilities to remain in the good or bad state, respectively.

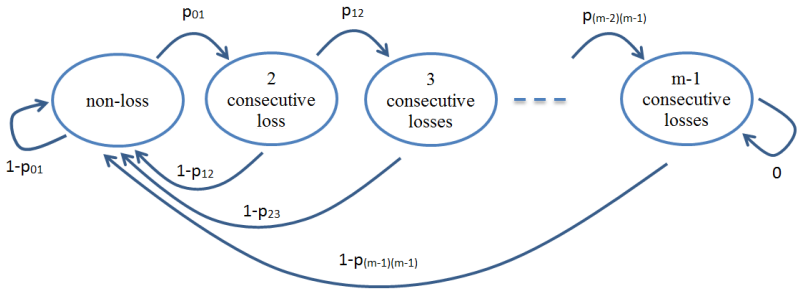



Figure 4.2: A $m$-th order extended Gilbert model

The model is depicted in Figure 4.1. Typically, it happens that $p + q < 1$: in the case $p + q = 1$, the Gilbert model reduces to the Bernulli one. In this work, the simplified version of the Gilbert-Elliot model is considered.

### 4.3.3 Extended Gilbert model

This Extended Gilbert model presents one good and $m-1$ bad states to generate a hyper-geometrical distribution of the duration of the good and the bad phases with specific transmission as approximation of a wireless channel. Unlike the general Markov model assuming all past events that might affect the future state, in an extended Gilbert model only the past $m$ consecutive packet loss will affect the future state. Figure 4.2 illustrates how the extended Gilbert model works. Such a model is an simplification of a general $M$-state model. A hyper-geometrical distribution is an useful approximation of loss burst with long term correlation, which are often observed on error prone wireless links.

## 4.4 Performance evaluation

The behavior of a centralized algorithm in presence of noise in the communication of individual wireless channels $a_{ij}$ is highlighted.

A swarm of $n = 6$ UAVs is considered in the simulations, resulting in a hexagonal configuration. The main task is to control the centroid of the swarm to follow a desired trajectory in an environment where some obstacles might get in the way. Hence, the second task is to keep the formation in a hexagonal configuration without violating the constraint to be far from such obstacles.

For the following simulations, a Gilbert model with six states has been employed. As shown by [52], the 3rd order Extended Gilbert Model model might be considered as a good approximation. The six states to correctly approximate the

actual pattern of the network and the values of the model transition matrix are derived from the experiments performed by [24].

In case no issues in the communication channels are present, the algorithm will successfully achieve the desired tasks. However, in the practice, the algorithm suffers from the loss of packets in the communication network due to high amount of data exchanged between the UAVs. Therefore, the control algorithm has been tested in various scenarios, in which the complexity of each scenario has been constantly increased so as to stress the algorithm and to bring to light the behavior of the agents.

### 4.4.1 Technical details

The swarm starts with a regular hexagon formation in which the distance of each agent from the center of the formation is equal to 1m. The initial position of the centroid of the swarm starts at the origin $x = y = z = 0$ and the final position is located at $x = 5$m. The coordinates $y, z$ of the swarm's centroid are kept the same, without loss of generality, for all the simulation The initial and final velocities of the swarm centroid are set to zero. The simulation lasts 100 seconds: in this time, the swarm's centroid has to move from the starting point to the end one along a planned linear trajectory. The controller sample time has been selected as 0.1s.

The natural frequencies and damping factor in (4.9) for each agent has been set to 5rad/s and 0.65, respectively. The distance $\rho_i$ in which each agent is able to detect obstacles has been set to 0.5m.

The tasks and the constraints employed in these simulations have been already introduced in Section 4.2.2. The following gains have been hence chosen for the two tasks $\boldsymbol{K}_1 = 30\boldsymbol{I}_3$ and $\boldsymbol{K}_2 = 5\boldsymbol{I}_{18}$, respectively, where $\boldsymbol{I}_\nu$ is a $(\nu \times \nu)$ identity matrix. Gain $k_\nabla$ in (4.3) has been set to 1. The threshold $\rho_s$ in (4.7) for the obstacle avoidance constraint has been selected equal to 1m, while $k_{i,k} = 20$, with $i = 1, \ldots, n$ and $k = 1, \ldots, o_i$.

In the following evaluation three main performance metrics will be considered, namely:

- *Maximum Agent Distance* (MAD). It is the maximum distance that an agent reaches from the center of the swarm.
- *Maximum Error* (ME). It is the maximum error from and agent to its reference value and equals to $\max_{i=1,\ldots,n} \|\boldsymbol{p}_{d,i} - \boldsymbol{p}_i\|$.
- *UAV collision.* This parameter is set to *yes* whether some collisions happen between the agents themselves or with an obstacle; otherwise it is set to *no.* Collisions are verified a posteriori by looking at the path of each agent. The effects of a collision on the agent's path after the impact have not been considered in the simulations.

Obviously, the employed loss models are probabilistic. Hence, the data that will be reported are just estimations of the average performance. However, this arrangement allows the identification of a threshold distance transmission, within which the UAVs are considered close enough to participate in the communication. Beyond this threshold, agents could be considered virtually lost. Hence, in order to emulate the loss of connectivity between an agent and the master, a node might be considered lost if it overcomes a certain distance from the swarm. Such a value has been set at $\sigma_t = 10$m in these simulations. If the master is lost, the mission is considered failed. Moreover, as an assumption, the master is never lost in the performed simulations.

Without loss of generality, in the simulations the UAVs move in a plane, i.e. $z = 0$. Moreover, the obstacles are considered as points in the space. Three scenario have been then considered in the following simulations. In the former, just one obstacle has been randomly put in the environment. In the second scenario, instead, a set of obstacles randomly get in the way during the movement of the swarm. In the latter, a corridor has been emulated through two walls made up of two straight lines full of obstacles separated themselves by very small distances: each agent of the swarm has to remain inside the corridor avoiding to be too close with respect to the walls, the others agents and other two obstacles.

### 4.4.2 Scenario 1. Single obstacle.

In case of ideal communication, the swarm perfectly behaves in presence of a single obstacle. The agents deviate from the desired path of a small amount necessary to avoid the obstacle. This behavior is summarized in Table 4.1. Already in this simple scenario, the insertion of loss models results in lower performance. This phenomenon is amplified especially in models with memory, in fact a prolonged loss of communication involves an accumulation of error that must be recovered from the controller.

| Model | MAD [m] | ME [m] | UAV Collision |
|---|---|---|---|
| Ideal | 0.41 | 1.32 | No |
| Bernulli | 0.42 | 1.28 | No |
| Gilbert | 0.67 | 1.35 | No |
| Gilbert n | 0.83 | 1.73 | No |

Table 4.1: Simulation results about Scenario 1.

### 4.4.3 Scenario 2. Cluster of obstacles.

In this second case study, the structure of the obstacles has been complicated with respect to the previous case. The presence of multiple obstacles at the same time

worsens performance. Table 4.2 summarizes the obtained results and shows that, in the presence of data loss, some agents might be considered lost since they are too far from the master. Some collisions may also happen.

Figure 4.3 shows the time histories of the distance of each agent from the center of the swarm, each color identifies an agent. The threshold $\sigma_t$ over than an agent is considered lost has been not depicted in this plot. In such a figure no package losses are considered in the network, hence no issues are revealed. Figure 4.4 shows a 3D plot of the trajectory for each agent. The coloured diamonds are the initial positions of each agent's path, while the coloured triangles are the arrival points end the Red diamonds are the considered obstacles. On the other hand, when simple Gilbert model for data loss is considered, some agents might exit the swarm, as depicted in Figure 4.5 where each color identifies an agent. A Gilbert model has been considered in such a case. The dashed red line represents the threshold $\sigma_t$..

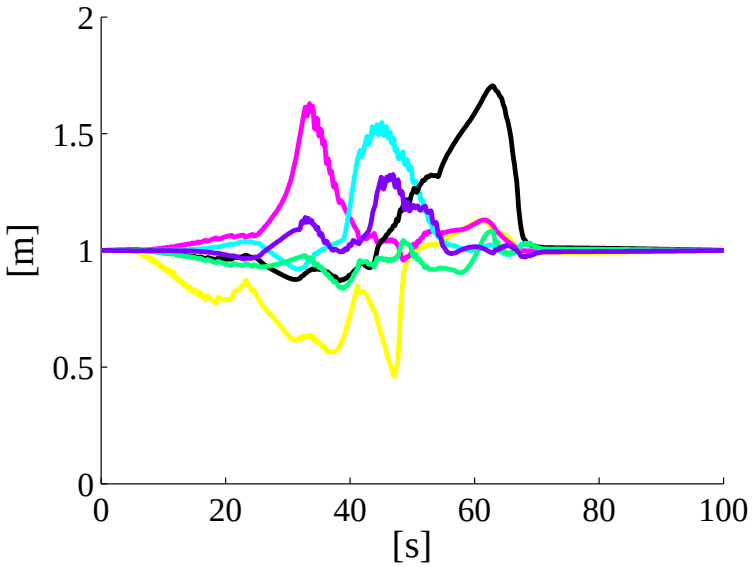



Figure 4.3: Time history about the distance of each agent from the center of the swarm.

### 4.4.4 Scenario 3. Corridor.

In this case the flight environment is different: a corridor has been simulated in which the swarm has to flight. Other two obstacles are placed inside this aisle.

The corridor's length is about 1m and it is passed through by the swarm of

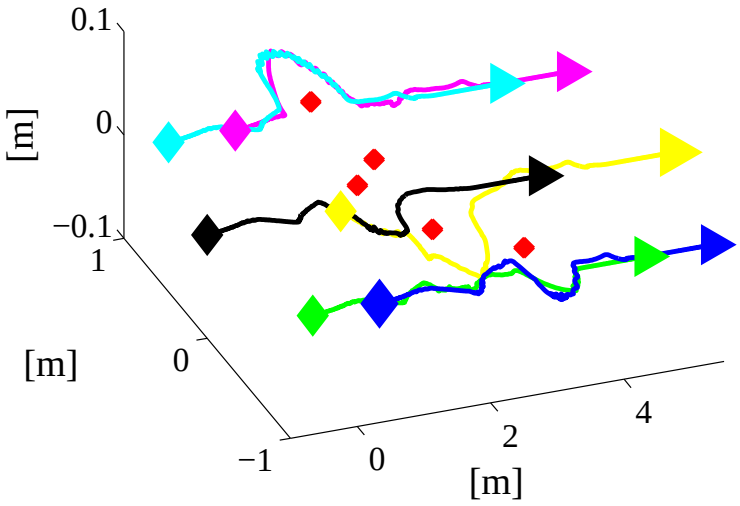



Figure 4.4: Trajectory for each agent in the 3D space.

| Model | MAD [m] | ME [m] | UAV Collision |
|---|---|---|---|
| Ideal | 1.07 | 1.69 | No |
| Bernulli | 1.34 | 2.30 | No |
| Gilbert | 20.12 | 19.23 | Yes |
| Gilbert n | 1.68 | 2.04 | No |

Table 4.2: Simulation results about Scenario 2.

UAVs after about 2 meters from the beginning of the task and thus when the platoon is already in a hexagonal formation.

The simulations, summarized in Table 4.3, show a clear difference between the case of perfect communication (see Fig. 4.6) and the implemented loss models. In figure each color identifies an agent. The coloured diamonds are the initial positions of each agent's path, while the coloured triangles are the arrival points. Red spots are the considered obstacles. Red continuous lines represent the walls of the considered corridor. From the bevelled path of certain UAVs, it is evident how the obstacle avoidance constraint acts when an agent passes between two obstacles inside the same security threshold $\rho_s$. Even with simple models, a fundamental dependence from data loss is evident.

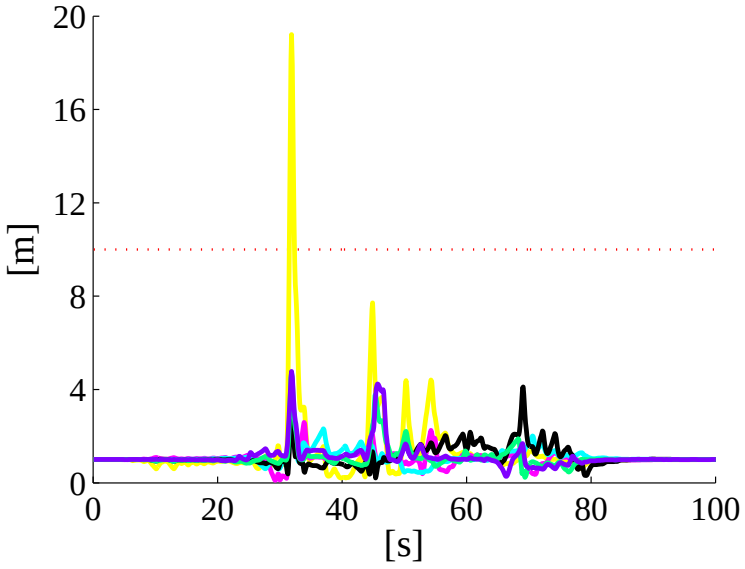



Figure 4.5: Time history about the distance of each agent from the center of the swarm.

| Model | MAD [m] | ME [m] | UAV Collision |
|---|---|---|---|
| Ideal | 0.47 | 1.03 | No |
| Bernulli | 53.2 | 53.35 | Yes |
| Gilbert | 197.06 | 197.9 | Yes |
| Gilbert n | 89.89 | 89.94 | Yes |

Table 4.3: Simulation results about Scenario 3.

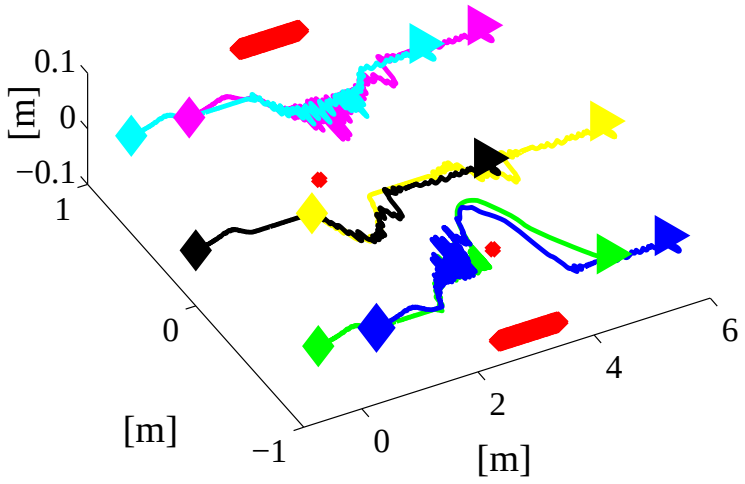



Figure 4.6: Trajectory for each agent in the 3D space.

# Chapter 5

# Conclusion and future research directions

A brief recap about the device end method presented in this thesis and the achieved results will be the object of the current chapter. Proposals for future research directions will be discussed as well.

## 5.1 Main results

A task about grasping and manipulation of objects with a multi-fingered robotic arm in unknown environments has been carried out during this thesis. The problem has been obviously split in three different parts, which are namely:

- Aerial manipulation with implementation of robotic arm for quad-rotor.
- Ground platform for cooperative operation of lending in difficult environment.
- Multi agent coordination for object manipulation in a cooperative way, exploiting the redundancy of the"robotic platform plus object".

For each of the previous items, a new device has been proposed and validated with several experiments (for the first two parts) and simulations for the last one.

The design of an ultra-lightweight robotic arm for aerial robotics applications has been presented. Mechanical design has been constrained by energy and payload limitations with the goal of maximizing the arm's payload and the displacement of the center of gravity. The kinematic and dynamic model have been studied, and validated through experimental measurements. The realization of the mechanical parts has been carried out using a 3D printer. FEM analysis has been conducted using CAD software to validate the mechanical design.

The inclusion of even simple loss models in centralized control algorithm to control agents' swarms leads to inevitably issues regarding the behavior of the agents and possible collisions between them. Although the performed analysis is just qualitative, due to the probabilistic nature of the loss models which generate random results, the manifestation of problems due to the loss of information in a network of agents coordinated by a central node has been highlighted. This is mainly due to the fact that the central node, e.g. the master, is subjected to a large amount of packages to manage, creating a bottleneck. Moreover, the possible loss of the master would lead to the conclusion of the mission.

## 5.2 Proposals for the future

Likewise performed in the previous section of this chapter, the directions for future researches can be also split in three parts.

Electronics have been designed following the same specifications used for the mechanics, constrained by weight and energy consumption limitations. Data communication between the electronic components has been obtained through an SPI bus that guarantees high data rates, while the communication between the arm and the UAV has been obtained through a custom protocol that lays upon the USB standard. The developed software libraries allow the user to control the arm using position, speed and torque references. The arm design has been experimentally validated through the implementation of position, velocity, and torque control schemes.

In future work, the degradation of communications due to the traffic of individual agents will be addressed, considering a model of loss which also contains the collisions due the hight number of agents. Moreover, the alienation of a single agent from the swarm will be considered: in such a case, this agent should perform an emergency landing, while the swarm should be reconfigured to keep the reference trajectory. Moreover, future plans will consider distributed algorithms.

# Appendix A

# Appendix A

As said at the beginning of Chapter 3, many other quality measures may be chosen in substitution of the proposed one in Chapter 3, without effecting the general proposed framework. The only thing to point out is that any other index should just ensure that it can be applied to discretized surfaces of the objects.

## A.1 Schematic

Since the reconstructed surface of the object is sampled by points/masses, the above method cannot be directly applied. Differently from the continuous case, due to the presence of a finite set of sampled points, the existence of a "grip plane" containing all the normal forces is not guaranteed. This is mainly due to the fact that, because of the discretization, the normals to the surface are an approximation of the real ones.

The next step consists in looking for an equilateral grasp configuration. To this aim, for each grasp configuration, the unit vector normal to the object's surface at each contact point is projected on the grip plane. Denoting with $\varphi_j$ the angle between these projections for each of the 3 couple of points of the considered

## A.2 PCB

The proposed quality index has been tested with simulations using synthesized objects.

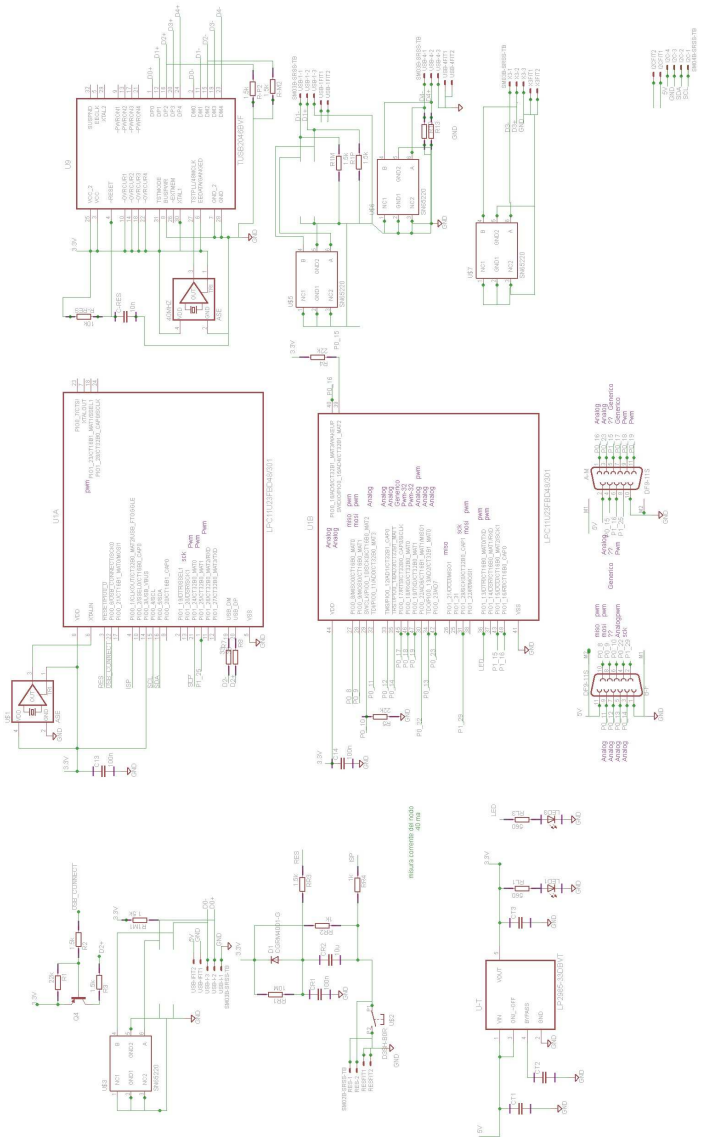


Figure A.1: Eagle schematic of the Control Node.

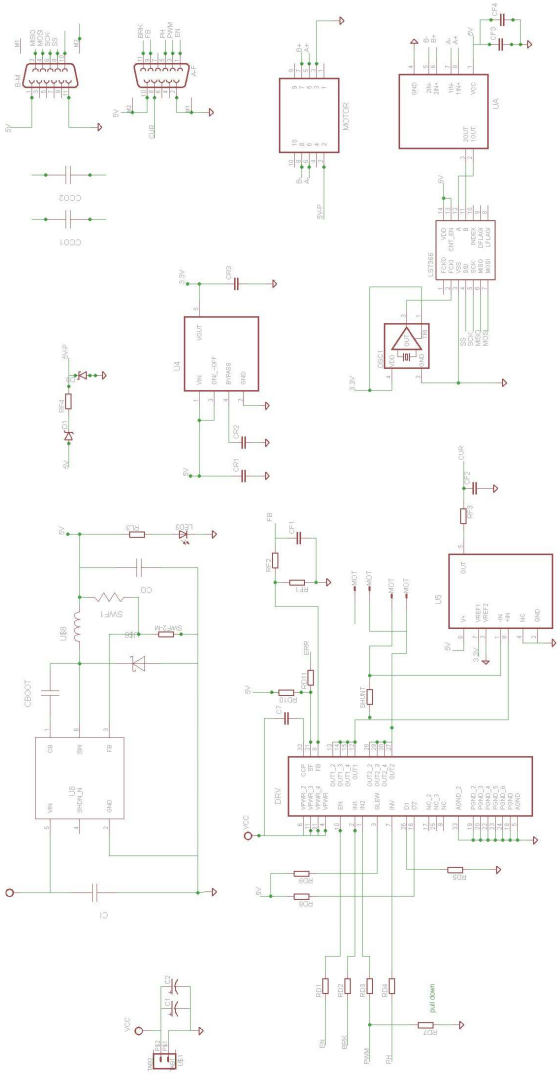


Figure A.2: Eagle schematic of power skin of master node.

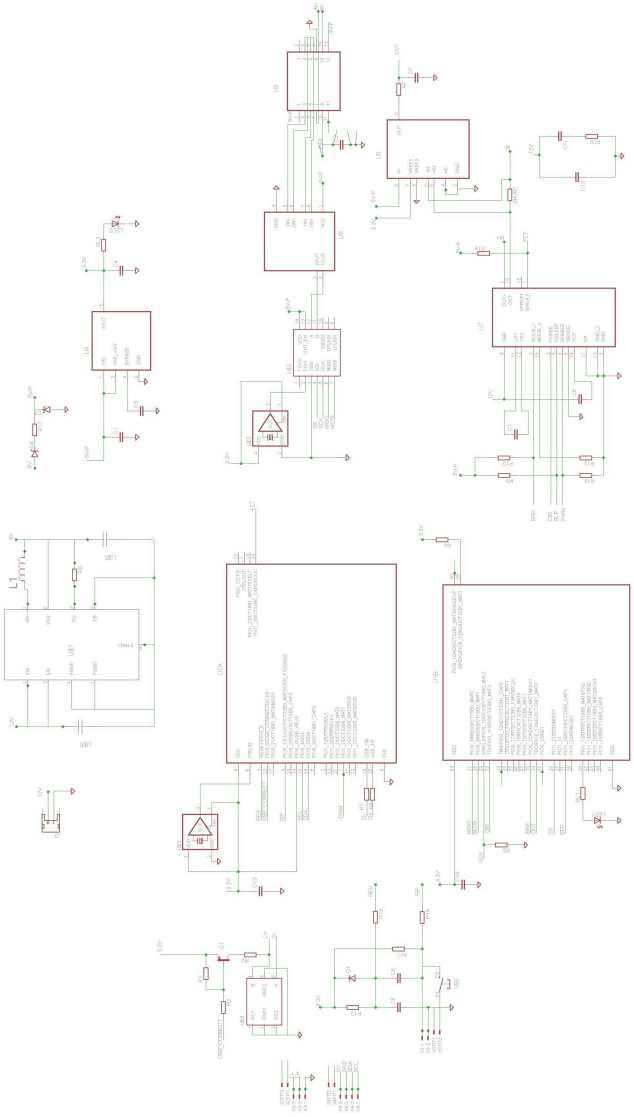


Figure A.3: Eagle schematic general servo node.

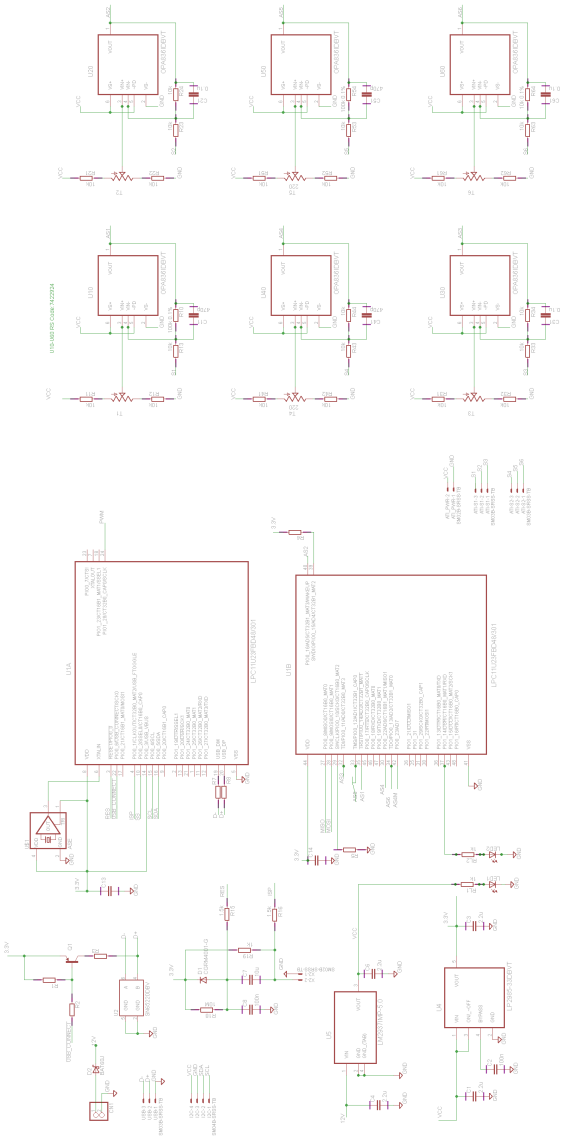


Figure A.4: Eagle schematic of ATI nano board.

# Appendix A

# Appendix B

In this appendix, this last statement will be proven by showing the procedure for the choice of the motor. The method proposed in [2] is hence adopted, suitably modified to cope configurations, assuming the static torque end the impulsive torque. A second version of robotic arm will be considered throughout this Appendix.

## A.1 Motor dimension procedure

Since the reconstructed surface of the object is sampled by points/masses, the above method cannot be directly applied. Differently from the continuous case, due to the presence of a finite set of sampled points, the existence of a "grip plane" containing all the normal forces is not guaranteed. This is mainly due to the fact that, because of the discretization, the normals to the surface are an approximation of the real ones.

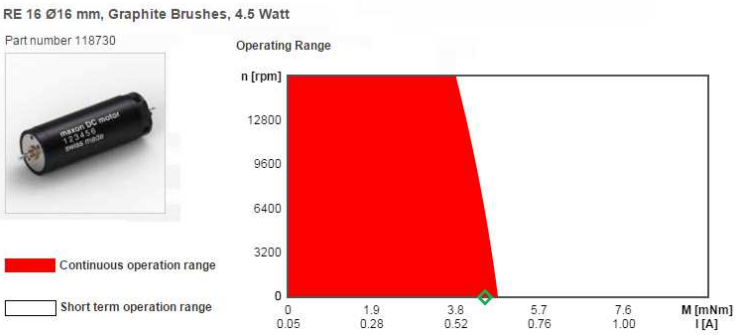



Figure A.1: Maxon motor working diagram.

## A.2 Motor project

The proposed quality index has been tested with simulations using synthesized objects.

The dynamic parameters of the reconstruction ellipsoid have been chosen as follows: $m = 10^{-3}\ kg$, $k = 0.3 \cdot 10^{-3}\ N/m$, $b = 0.09 \cdot 10^{-3}\ Ns/m$, and $F_a = 5\ N$.

The next step consists in looking for an equilateral grasp configuration. To this aim, for each grasp configuration, the unit vector normal to the object's surface at each contact point is projected on the grip plane. Denoting with $\varphi_j$ the angle between these projections for each of the 3 couple of points of the considered

| | |
|---|---|
| **Values at nominal voltage** | |
| Nominal voltage | 12 V |
| No load speed | 13900 rpm |
| No load current | 45.9 mA |
| Nominal speed | 11900 rpm |
| Nominal torque (max. continuous torque) | 4.36 mNm |
| Nominal current (max. continuous current) | 0.582 A |
| Stall torque | 34.3 mNm |
| Stall current | 4.23 A |
| Max. efficiency | 79 % |
| **Characteristics** | |
| Terminal resistance | 2.83 Ω |
| Terminal inductance | 0.113 mH |
| Torque constant | 8.11 mNm/A |
| Speed constant | 1180 rpm/V |
| Speed / torque gradient | 411 rpm/mNm |
| Mechanical time constant | 5.78 ms |
| Rotor inertia | 1.34 gcm² |
| **Thermal data** | |
| Thermal resistance housing-ambient | 30 K/W |
| Thermal resistance winding-housing | 8.5 K/W |
| Thermal time constant winding | 10.9 s |
| Thermal time constant motor | 504 s |
| Ambient temperature | -20...+65 °C |
| Max. winding temperature | +85 °C |
| **Mechanical data** | |
| Bearing type | sleeve bearing |
| Max. speed | 16000 rpm |
| Axial play | 0.05 - 0.15 mm |
| Radial play | 0.014 mm |
| Max. axial load (dynamic) | 0.8 N |
| Max. force for press fits (static) | 15 N |
| (static, shaft supported) | 0 N |
| Max. radial load | 1.5 N, 5 mm from flange |
| **Other specifications** | |
| Number of pole pairs | 1 |
| Number of commutator segments | 7 |
| Number of autoclave cycles | 0 |
| **Product** | |
| Weight | 40 g |

Figure A.2: Maxon motor data.

Figure A.3: Motor dimension example.

# Bibliography


[1] *Ascending technologies, GmbH.*

[2] *MAXON Motors: development motor ed tool for robotic devices.*

[3] *NXP MBED: development platform for devices.*

[4] G. Antonelli, *Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems*, IEEE Transactions on Robotics **25** (2009), no. 5, 985–994.

[5] G. Antonelli, *Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotics systems*, IEEE Transactions on Robotics **25** (2009), no. 5, 985–994.

[6] G. Antonelli, F. Arrichiello, and S. Chiaverini, *The null-space-based behavioral control for autonomous robotic systems*, Journal of Intelligent Service Robotics **1** (2008), no. 1, 27–39.

[7] G. Arleo, F. Caccavale, G. Muscio, and F. Pierri, *Control of quadrotor aerial vehicles equipped with a robotic arm*, 21st Mediterranean Conference on Control Automation, 2013, pp. 1174–1180.

[8] A. A. Belhoul, Y. A. Sekercioglu, and N. Mani, *Mobility protocols and RSVP performance in wireless IPv6 networks: Shortcomings and solutions*, Wireless Communications and Mobile Computing **8** (2007), no. 9, 1183–1199.

[9] C. D. Bellicoso and M. Caputano, $DAMA^{ROB}$, DArio and MArco's ROBotics symbolic toolbox for MATLAB.

[10] A. Bicchi, A. Danesi, G. Dini, S. La Porta, L. Pallottino, I.M. Savino, and R. Schiavi, *Heterogeneous wireless multirobot system*, IEEE Robotics and Automation Magazine **15** (2008), no. 1, 62–70.

[11] F. Caccavale, V. Lippiello, G. Muscio, F. Pierri, F. Ruggiero, and L. Villani, *Kinematic control with force feedback for a redundant bimanual manipulation system*, 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (San Francisco, CA), 2011, pp. 4194–4200.

[12] ———, *Grasp planning and parallel control of a redundant dual-arm/hand manipulation system*, Robotica **31** (2013), no. 7, 1169–1195.

[13] C. Costanzo, V. Loscri, E. Natalizio, and T. Razafindralambo, *Nodes self-deployment for coverage maximization in mobile robot networks using an evolving neural network*, Computer Communications **35** (2011), no. 9, 1047–1055.

[14] S. Dixit, E. Yanmaz, and O.K. Tonguz, *On the design of self-organized cellular wireless networks*, IEEE Communications Magazine, Feature Topic: Advances in Self-Organizing Networks **43** (2005), no. 7, 86–93.

[15] C.E. Doyle, J.J. Bird, T.A Isom, C.J. Johnson, J.C. Kallman, J.A Simpson, R.J. King, J.J. Abbott, and M.A Minor, *Avian-inspired passive perching mechanism for robotic rotorcraft*, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011, pp. 4975–4980.

[16] C.E. Doyle, J.J. Bird, T.A Isom, J.C. Kallman, D.F. Bareiss, D.J. Dunlop, R.J. King, J.J. Abbott, and M.A Minor, *An avian-inspired passive mechanism for quadrotor perching*, IEEE/ASME Transactions on Mechatronics **18** (2013), no. 2, 506–517.

[17] P.M.M. Fliess, J. Levine, and P. Rouchon, *Flatness and defect of nonlinear systems: Introductory theory and examples*, International Journal of Control **61** (1995), no. 6, 1327–1361.

[18] F. Forte, R. Naldi, A. Macchelli, and L. Marconi, *Impedance control of an aerial manipulator*, American Control Conference, 2012, pp. 3839–3844.

[19] J.A. Guerrero, Y. Challal, and P. Castillo, *Impact of wireless medium access protocol on the quadrotor formation control*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2012.

[20] Shi J.-Simmons R. Singh S. Hammer B., Koterba S., *An autonomus mobile manipulator for assembly tasks*, IEEE Xplore (2009).

[21] O. Hohlfeld, R. Geib, and G. Hasslinger, *Packet loss in real-time services: Markovian models generating QoE impairments*, 16th International Workshop on Quality of Service (Enschede, NL), 2008, pp. 239–248.

[22] F. Huber, K. Kondak, K. Krieger, D. Sommer, M. Schwarzbach, M. Laiacker, I Kossyk, S. Parusel, S. Haddadin, and A Albu-Schaffer, *First analysis and experiments in aerial manipulation using fully actuated redundant robot arm*, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 3452–3457.

[23] Petrucelly V. J., *Resistance wire strain gauges as elements of the wheatstone bridge*, AIEE Transactions (2010).

[24] Sheng-Dong-Xu S. Jia Y., Song X., *Modelling and motion analysis of fouf-mecanum omni-directional mobile platform*, IEEE Xplore (2013).

[25] W. Jiang and H. Schulzrinne, *Modelling of packet loss and delay and their effect on real-time multimedia service quality*, 10th International Workshop on Network and Operating System Support for Digital Audio and Video (Chapel Hill, NC, USA), 2000, pp. 1–9.

[26] A. E. Jimenez-Cano, J. Martin, Guillermo Heredia, Anbal Ollero, and R. Cano, *Control of an aerial robot with multi-link arm for assembly tasks.*, IEEE International Conference on Robotics and Automation, 2013, pp. 4916–4921.

[27] Ryan C. Julian, Cameron J. Rose, Humphrey Hu, and Ronald S. Fearing, *Cooperative control and modeling for narrow passage traversal with an ornithopter MAV and lightweight ground station*, International Conference on Autonomous Agents and Multi-agent Systems, 2013, pp. 103–110.

[28] Torres-Jara E. Kemp C., Edsinger A., *Challenges for robotr manipulation in human environments*, IEEE Xplore (2007).

[29] Sung-Hwan Kim, Young June Shin, Kyung-Soo Kim, and Soohyun Kim, *Design and control of robot manipulator with a distributed actuation mechanism*, Mechatronics **24** (2014), no. 8, 1223 – 1230.

[30] Lee C. Kwon S. Kim Yongdae, Kim Youngdeok, *Thin polysilicon gauge for strain measurement of structural elements*, IEEE Xplore (2010).

[31] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar, *Construction of cubic structures with quadrotor reams*, Proceedings of Robotics: Science and Systems, 2011.

[32] V. Lippiello and F. Ruggiero, *Cartesian impedance control of a UAV with a robotic arm*, 10th International IFAC Symposium on Robot Control, 2012.

[33] ———, *Exploiting redundancy in cartesian impedance control of UAVs equipped with a robotic arm*, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 3768–3773.

[34] V. Lippiello, F. Ruggiero, and L. Villani, *Exploiting redundancy in closed-loop inverse kinematics for dexterous object manipulation*, IEEE International Conference on Advanced Robotics (Munich, D), 2009.

[35] Stefanescu D. M., *Strain gauge and wheatstone bridges basic instrumentation and new applications for electrical measurement on non-electrical quantities*, IEEE Xplore (2011).

[36] S. Manfredi, *An algorithm for fast rendezvous seeking of wireless networked robotic systems*, Ad Hoc Networks **11** (2013), no. 7, 1942–1950.

[37] ———, *Design of a multi-hop dynamic consensus algorithm over wireless sensor networks*, Control Engineering Practice **21** (2013), no. 4, 381–394.

[38] Beetz M. Pangercic D. Siegwart R. Lutenegger S. Carloni R. Stramigioli S. Bruyninckx H. Doherty P. Siciliano B. Sala A. Tomatis N. Marconi L., Melchiorri C., *The sherpa project: smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments*, IEEE Xplore (2013).

[39] G.L. Mariottini, F. Morbidi, D. Prattichizzo, N. Vander Valk, G.J. Pappas, and K. Daniilidis, *Vision-based localization for leader-follower formation control*, IEEE Transactions on Robotics **25** (2009), no. 6, 1431–1438.

[40] F. Morbidi, R.A. Freeman, and K.M. Lynch, *Estimation and control of UAV swarms for distributed monitoring tasks*, Proceedings of the American Control Conference (San Francisco, CA), 2011, pp. 1069–1075.

[41] K. Nonami, F. Kendoul, S. Suzuki, and W. Wang, *Autonomous flying robots. unmanned aerial vehicles and micro aerial vehicles*, Springer-Verlag, Berlin Heidelberg, D, 2010.

[42] K. Nonami, F. Kendoul, S. Suzuki, W. Wang, and D. Nakazawa, *Autonomous flying robots*, Springer, 2010.

[43] M. Orsag, C. Korpela, S. Bogdan, and P. Oh, *Valve turning using a dual-arm aerial manipulator*, International Conference on Unmanned Aircraft Systems, 2014, pp. 836–841.

[44] L.E. Parker, *Multiple mobile robot systems*, Springer Handbook of Robotics (B. Siciliano and O. Khatib, eds.), Springer, 2008, pp. 921–941.

[45] Paul E. Pounds, Daniel R. Bersak, and Aaron M. Dollar, *Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control*, Autonomous Robots **33** (2012), no. 1-2, 129–142.

[46] PaulE. Pounds and AaronM. Dollar, *Aerial grasping from a helicopter UAV platform*, Experimental Robotics (Oussama Khatib, Vijay Kumar, and Gaurav Sukhatme, eds.), Springer Tracts in Advanced Robotics, vol. 79, Springer Berlin Heidelberg, 2014, pp. 269–283.

[47] P.E.I. Pounds and A.M. Dollar, *UAV rotorcraft in compliant contact: stability analysis and simulation*, IEEE/RSJ International Conference on Intelligent Robots and Systems (2011), 2660–2667.

[48] F. Ruggiero, J. Cacace, H. Sadeghian, and V. Lippiello, *Impedance control of VToL UAVs with a momentum-based external generalized forces estimator*, IEEE International Conference on Robotics and Automation, 2014.

[49] B. Siciliano and O. Khatib, *Springer handbook of robotics*, Springer, 2009.

[50] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics. modelling, planning and control*, Springer, London, 2008.

[51] ———, *Robotics: Modeling, planning and control*, Springer, 2009.

[52] C. Tang and P.K. McKinley, *Modeling multicast packet losses in wireless LANs*, Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems (New York, USA), 2003, pp. 130–133.

[53] Mandru D. Ardelean I. Plesa A. Tatar M., Popovici C., *Design and development of an autonomus omni-directional mobile robot vith mecanum wheels*, IEEE Xplore (2014).

[54] Doroftei I. Veaceslav S., Grosu V., *Modelling and control of an omni-directional vehicle with nmecanum wheels*, IEEE Xplore (2007).

[55] Mori Shunji Wada Masayoshi, Tominaga Yasutaka, *Omnidirectional holonomic mobile robot using nonholonomic wheels*, IEEE Xplore (1995).

[56] Y. Xunq, Modestino J.W., and X. Tian, *The accuracy of Gilbert model in predicting packet-loss statistic for single-multiplexer network model*, 24th Annual Joint Conference of the IEEE Computer and Communications Societies (Miami, FL, USA), vol. 4, 2005, pp. 2602–2612.

[57] Lelai Zhou, Shaoping Bai, and Michael Rygaard Hansen, *Design optimization on the drive train of a light-weight robotic arm*, Mechatronics **21** (2011), no. 3, 560–569.